ESD-TR-79-211

LEVEL

12

AD A082395

Semiannual Technical Summary

Distributed Sensor Networks

31 March 1979

Prepared for the Defense Advanced Research Projects Agency under Electronic Systems Division Contract F19628-78-C-0002 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

Approved for public release; distribution unlimited.

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract F19628-78-C-0002 (ARPA Order 3345).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Raymond L. Loiselle

Raymond L. Loiselle, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

11 31 Mar 79

6 DISTRIBUTED SENSOR NETWORKS,

12 58

9 SEMIANNUAL TECHNICAL SUMMARY REPORT, 1 Oct 78 – 31 Mar 79,

TO THE

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

10 Richard T. Lacoss

15 F19628-78-C-ØØØ2, ARPA Order-3345

16 9D3Ø

1 OCTOBER 1978 – 31 MARCH 1979

18 ESD

ISSUED 4 FEBRUARY 1980

19 TR-79-211

Approved for public release; distribution unlimited.

LEXINGTON MASSACHUSETTS

207650

mit

ABSTRACT

Progress on the Distributed Sensor Networks program is reported. Further design and analysis of strawman sensor and report circuit communications for a low-flying aircraft DSN are presented. System studies dealing with sensor redundancy, adaptive communications, nonuniform node distribution, and alternative DSN applications are summarized. Overall designs for DSN node software, for a software development DSN simulation test bed, and acoustic node front-end simulation software are reviewed. Progress on development of a digital acoustic data collection system is reported.

CONTENTS

SUMMARY

This fourth Semiannual Technical Summary (SATS) for the Distributed Sensor Networks (DSN) program reports research results for the period 1 October 1978 through 31 March 1979. Topics covered include analysis of communication circuits of the strawman design described in the previous SATS, several smaller DSN system studies, software definition and development, and experimental facilities.

The strawman system design presented in the previous SATS included sensor circuits for single-hop local exchange of sensor information and report circuits to deliver surveillance information to sector nodes. The performance of those circuits has been studied analytically and using Monte Carlo methods to determine performance characteristics. The circuits are time-slotted round robin. Node distribution, communication connectivity, and single-try single-hop message reception success were as described in the previous SATS. The circuits have been demonstrated to effectively perform their assigned functions even when substantial numbers of system nodes were not functional. In our design the sensor and report circuits carry the bulk of the routine traffic and our analysis has confirmed that we should have no fundamental communication difficulties. First versions of strawman packet and message formats for the sensor and report circuits are reviewed in Sec. I which also contains all the communication circuit analysis.

Section II contains the results of four system studies which have been completed. One Monte Carlo study quantitatively demonstrates the intuitively obvious advantage of sensor redundancy in a DSN. A second study demonstrates the importance and advantage of adaptive communication algorithms in a DSN. The third analysis addresses the issue of the cost effectiveness and utility of a DSN with nonuniform sensor densities. The last topic covered is a strawman DSN solution to a hostile weapon location problem.

Progress in the DSN software area is reported in Sec. III. A first-version organization for the software for DSN nodes is described in some detail. That organization is appropriate for the strawman system described in our previous SATS and for a three-node system with acoustic sensors which is being planned for FY 80. Progress on a software test bed and general software tools required for continuing DSN research and system development is reviewed. A simulation package which will allow us to realistically simulate acoustic DSNs without using real data or very large amounts of computer time to simulate the computationally intensive signal processing at each node is also reported.

Finally, progress is reported on the development of a digital data acquisition system which will be used to further develop and evaluate small acoustic array capabilities and will be the basis for developing an experimental DSN node.

CONTRIBUTORS
TO
DISTRIBUTED SENSOR NETWORKS PROGRAM

GROUP 22

Lacoss, R. T.

Demko, P., Jr.
Green, P. E.
Landers, T. E.
Retzlaff, A. T.
Walton, R. L.

DISTRIBUTED SENSOR NETWORKS

I. STRAWMAN SYSTEM ANALYSIS

Basic performance and reliability analysis for two of the communication circuits of the strawman DSN described in our 30 September 1978 Semiannual Technical Summary† has been completed. The two circuits represent unsophisticated versions of a slotted round robin approach to providing essential DSN communications under circumstances of continuous channel overload. As discussed in the previous SATS the communication scheme is quite simple. It may be that DSN communications can be less structured and/or more sophisticated than the circuits analyzed here. Such alternatives, which would have been much more difficult to evaluate in the short term, will be considered in the ongoing research. In any case, from the analysis of the simple strawman circuits, we now have confidence that communication should not be a fundamental limitation for a low-flying aircraft surveillance DSN.

In addition to the analysis, further detailed design of the strawman circuits has been done and is reported here. In general, the more detailed design was carried at least to the point required to undertake meaningful analysis. It also represents a step toward first versions of actual DSN software.

DSN does require communication in addition to that provided by the circuits discussed here. For example, communication involved in system control is very important. However, that communication should be relatively low rate on average although it may be bursty. It should not stress the communication channel and can be provided for by small additions to the mechanisms described here or by small changes to standard packet radio usage and protocols. Such control communication is not addressed here since it should not be the source of any fundamental system problems resulting from limited channel capacity.

The geographical deployment of the strawman DSN and a number of assumed basic communication parameters were presented in our September 1978 SATS. For convenience we reiterate some of that information here.

We assume that all nodes are located on a hexagonal grid on 10-km centers (Fig. I-1). A link is a pair of nodes. The length of a link is the distance between the pair of nodes. However, not all links are usable for communication in the DSN. For example, terrain will block many links and others will be too long for communication. We assume that 70 percent of all 10-km links and 50 percent of all 17-km links are usable. All links longer than 17 km are assumed to be unusable. We assume that there will be 30 percent errors on usable links. Such errors can be attributed to fading, multipath, noise, collisions, etc. For simplicity we also assume that all usable links are symmetric so the probability of successful reception for packets going one way on a link is the same as that for packets going the other way. These assumptions appear to be reasonable for well-sited packet radios but probably not for radios whose deployment has not been optimized in some way.

The following two sections assume the above deployment and communication parameters and present the result of analysis of the strawman DSN sensor and report circuits. These are

† Distributed Sensor Networks SATS, Lincoln Laboratory, M.I.T. (30 September 1978), DDC AD-A070455.

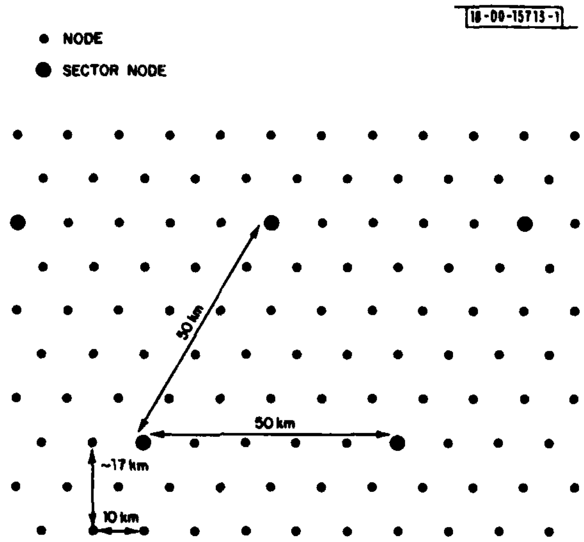


Fig. I-1. Geographical distribution of strawman DSN system nodes.

the basic circuits used to locally exchange sensor information (sensor circuit) between nodes and to transfer surveillance reports within a sector to a sector node (report circuit).

A. THE SENSOR CIRCUIT

In the sensor circuit, every node broadcasts a packet once each second. There is no required order to the broadcasts. Broadcasts are expected to reach the node's 12 nearest neighbors, 6 nodes 10 km distant, and 6 nodes 17 km distant, whenever the links to these nodes are usable. On the average, 6×70 percent = 4.2 10-km links and 6×50 percent = 3.0 17-km links will be usable. Data received by a node on the sensor circuit are not repeated by the receiving node. All sensor circuit communication is one hop.

1. Sensor Circuit Communication Defects

Analysis has been completed to determine whether the sensor circuit will provide adequate communication for the purpose of tracking targets. To do this, sensor data from nodes must be received at other nodes which can perform tracking. Since any node can do tracking, the important question is whether there is any node that gets all the sensor data from some set of nodes that can be expected to see a target. For example, given two nodes 10 km apart, is there any node that gets the sensor circuit data from both those nodes? If not, the DSN has a sensor communication defect.

The percentage of sensor communication defects in a DSN will be very much a function of the percentage of functional nodes in the DSN. As nodes stop functioning, either for lack of maintenance or as attack casualties, the sensor communication defects will increase. However, such defects may still be a small fraction of the difficulties caused directly by nonfunctioning nodes. If either member of a node pair does not function at all then data from that pair cannot be jointly used for tracking. This is independent of any communication problems and is termed

a functional defect. For example, if a random 10 percent of all nodes do not work, then 19 percent of all pairs of nodes 10 km apart will have at least one nonworking member and be instances of functional defects.

The study reported here confirms in general that sensor communication defects will be much less significant than functional defects. Of course this conclusion is for the specific situations and communication parameter values used.

Figure I-2 indicates the possible tracking nodes for a single node pair. There are eight such possible tracking nodes including the two nodes of the pair themselves. The probabilities that each node will be a tracking node are also shown in the figure. All these probabilities are independent except that either both or neither of the two nodes in the pair will be tracking nodes. The information on this figure can be used to compute the probabilities of a node pair having functional or sensor communication defects. These have been computed and are given in Table I-1. The table shows that sensor communication defects are infrequent compared to functional defects for all situations other than nearly 100 percent nodes functional, in which case neither kind of defect is very significant.

The concept of functional and sensor communication defects can be extended to situations other than two-node tracking with nodes separated by 10 km. Suppose there are three nodes at the corners of a triangle with 10-km sides. If one or more of these nodes is not functional the triangle has a three-node functional defect. If all three nodes are functional but the required sensor circuit messages from all three do not get to a node which can jointly use them to track targets then the node triangle has a three-node sensor communication defect. The number of three-node defects (see Fig. I-3) is given in Table I-2. Again the frequency of sensor communication defects is small compared with the frequency of functionality defects.

Other variants of sensor circuit functional and communication defects have also been studied with similar results.

2. Sensor Circuit Packet and Message Formats

Sensor circuit packets are composed of sensor messages. Strawman versions of sensor circuit packet and message formats have been designed in some detail. This is a first step toward detailed designs which can be used in future software development. Without presenting the formats we summarize here some of the characteristics of the packets and messages of this strawman design. Packet radio synchronization, error detection and correction, and other header bits are in addition to the data bits which are discussed here.

Each sensor circuit packet contains one identification-synchronization message and a number of sensor messages. The entire packet contains information from and about a single DSN node.

The identification-synchronization message in a packet contains bits to indicate that it is an identification-synchronization message, to identify the source node, and to specify absolute reference times for acoustic and radar measurements contained in the sensor messages within the same packet. The use of reference times allows the sensor messages to be more compact. As presently defined the identification-synchronization message consists of 68 bits.

Besides a synchronization message, each sensor packet contains a number of sensor messages, each describing one possible target. There are two types of sensor messages: acoustic sensor messages for acoustic data, and radar sensor messages for radar data. Both types of

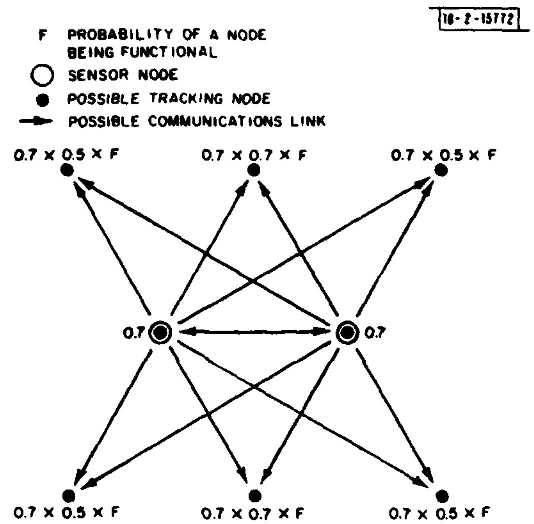


Fig. I-2. Eight possible tracking nodes for a single pair of sensor nodes. Probability of being a tracking node is indicated next to each of the eight nodes.

TABLE I-1

PROBABILITY OF A PAIR OF NODES 10 km APART HAVING A FUNCTIONAL OR SENSOR COMMUNICATION TRACKING DEFECT WITH 70-PERCENT 10-km AND 50-PERCENT 17-km LINKS USABLE

Percentage of Functional Nodes	Probability of a Two-Node Defect	
	Functional	Communication
100	0.00	0.014
90	0.19	0.017
80	0.36	0.019
70	0.51	0.021
60	0.64	0.021
50	0.75	0.020
40	0.84	0.017
30	0.91	0.013
20	0.96	0.007
10	0.99	0.002

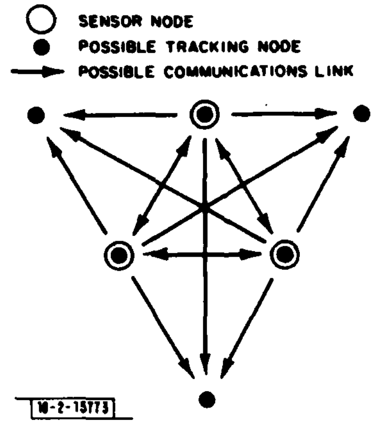


Fig. I-3. Six possible tracking nodes for a triangle of sensors with 10-km separations.

TABLE I-2

PROBABILITY OF A TRIANGLE OF NODES 10 km APART HAVING A FUNCTIONAL OR SENSOR COMMUNICATION TRACKING DEFECT WITH 70-PERCENT 10-km AND 50-PERCENT 17-km LINKS USABLE

Percentage of Functional Nodes	Probability of a Three-Node Defect	
	Functional	Communication
100	0.00	0.093
90	0.27	0.075
80	0.49	0.057
70	0.66	0.042
60	0.78	0.029
50	0.88	0.018
40	0.94	0.010
30	0.97	0.005
20	0.99	0.001
10	0.99	0.000

messages start with a 20-bit header which identifies the message type, relative measurement time, and a target identification tag assigned by the data source node.

If the message is an acoustic sensor message the strawman format specifies that it also contains a most recent azimuth measurement for the target, the time interval between azimuth measurements, up to four past azimuth measurements, and detailed spectral characterization for the most recent measurement. The format requires some 60 bits for spectral characterization and from 15 to 39 bits for the other information. Thus the acoustic sensor message sizes range from some 95 to 119 bits. A single sensor circuit packet with 1500 data bits, as assumed for the strawman design reported in the September 1978 SATS, could hold on the order of 10 such sensor messages in addition to an identification-synchronization message.

At this time a detailed strawman format for radar reports has not been designed. However, it should be similar to the acoustic case in that each target report will correspond to a message and a message will contain on the order of 100 bits. Both acoustic and radar sensor messages can be included in a single sensor packet. If both exist they must share the total available bits so, for example, a packet might contain 5 sensor reports of each type.

Finally we should note that the amount of redundantly transmitted information in the report circuit can range from about 100 percent to only a very little. New data are obtained from sensors only once every 2 sec in the strawman design and sensor circuit packet broadcasts are made once each second. If all sensor messages from the 2 sec fit into a single packet they would be repeated completely once before new data become available. However, if both radar and acoustics are in use and/or there are (or appear to be) many possible targets, all messages for a 2-sec interval may not fit into a single packet. In this case two packets would be used with only a small amount of important information repeated, as the message format design allows for. If the number of messages per observation interval increased further, the time between observations would be increased to accommodate the messages in the available communication capacity. Such variation of the effective broadcast sensor measurement interval can be done on an individual target basis with slowly changing or well-established targets reported less often.

3. Data Retransmission

The sensor circuit defect results reported in Sec. A-1 above assumed that sensor data could be communicated over any usable communication link. To do so, some data must be rebroadcast. Some simple results related to such retransmission are presented in this section. The analysis here is limited to the case when all DSN nodes are functional.

If a datum is broadcast for N sec in a row, its probability of failing to arrive at a given neighboring node with a usable link within N sec is given in Table I-3. Thus, if an important piece of information is broadcast 4 times in a row it will usually (99 times out of 100) get to a given neighboring node with a usable link. This simple table just indicates the improvement in reliability which can be obtained if important azimuth information is repeated on the sensor circuit.

However on the average, each node has usable links to 7.2 other nodes. Consider a node with usable links to 7 other nodes. Suppose that for some reason we want a datum to be received by all 7 nodes with higher probability than would be obtained by a single transmission. Table I-4 gives the probability that more than N broadcasts will be needed to get the datum to

TABLE I-3

PROBABILITY OF DATUM FAILING TO GET FROM ONE NODE TO ANOTHER AFTER N sec

N sec	Probability of Failure
1	0.3
2	0.09
3	0.027
4	0.0081
5	0.0024
6	0.00073

TABLE I-4

PROBABILITY OF AT LEAST ONE OF 7 NODES FAILING TO GET DATUM AFTER N sec WITH 0.70 PROBABILITY OF SUCCESS FOR INDIVIDUAL PACKET RECEPTION

N sec	Probability of Failure
1	0.92
2	0.48
3	0.17
4	0.055
5	0.017
6	0.0051

all 7 neighbors. The failure rate is very high for a single broadcast and goes quickly to only about 5 percent for 4 transmissions.

If retransmission is used to improve reliability, as it is to some extent in the strawman system, a scheme involving positive acknowledgments might be considered to reduce channel utilization. For example, from Table I-4 we note that if acknowledgments are used to control retransmission and the number of retransmissions is limited to 3, then the average number of retransmissions would be $0.92 + 0.48 + 0.17 = 1.57$. On the other hand, if we do not use acknowledgments, but simple retransmit data only 3 times, we will transmit that data $3 - 1.57 = 1.43$ more times. However, this ignores the communication overhead to implement positive acknowledgment and that could be considerable. Of course, if lower probability of failure were required the potential advantage of acknowledgments becomes greater. However, we require only that a message get to several sites with very high probability, not to all sites.

The trade-off to be considered is the simplicity of the strawman open-loop approach vs some potential communication capacity conservation if a complicated multi-destination acknowledgment and retransmission scheme could be developed. Our belief is that the potential gain of the more complicated scheme is not currently substantial enough to justify its development, particularly since it appears the simple scheme is adequate for the task. Of course this follows largely from the fact that the sensor circuit involves only a multi-destination single-hop service and can tolerate reasonable amounts of lost packets due to inherent information redundancy.

B. THE REPORT CIRCUIT

In the DSN of Fig. I-1 there are special sector nodes every 50 km. These nodes can be considered as gateways to other systems or representative of local users or command centers in the DSN. In any case, detection and tracking information must flow from arbitrary DSN nodes toward the sector nodes, and control information must flow in the reverse direction. The report circuit described in our September 1978 SATS was to perform such functions. In particular, it was to cause target reports within the DSN to flow to the nearest sector node. Some analysis of the characteristics of that circuit are reported here.

For convenience we begin by reviewing the report circuit design and adding details as needed. This description of the report circuit and the analysis are limited to the flow of report information to the nearest sector node. We do not deal with more complicated reporting functions or with control information which will be lower rate and will probably require higher level protocols. Also, it should be noted that the report circuit as considered here is limited in scope. More sophisticated reporting requirements do exist. The definition of those requirements and the way to meet them are not within the scope of the current effort.

In the report circuit each node broadcasts once every second, just as in the sensor circuit. However, a major difference is that the report circuit in general is for multiple-hop data transfer. This multiple-hop transfer to the sector node is accomplished as follows. Each node broadcasts a target surveillance report once each second. That report incorporates all information available to the node at the time of the broadcast. It is the best world view that this node can generate. Of course every node cannot transmit information about every target in the network. If they did, a very large network would saturate with a very small number of targets. The simplest strategy is to have a node repeat information about every target within some range of the node. Under normal condition the strawman design specified this range to be 30 km. This means that target information will reliably propagate roughly 30 km in all directions and is in fact

selected to achieve condensation of reports at sector nodes which are separated from each other by some 50 km. This strategy has the advantage that it does not bias the network too strongly in favor of sector nodes. Any node in the network will have an up-to-date report on all targets within 30 km. If a sector node does fail, the reporting geometry – the size and shape of the region in which a target must lie before reports about it are repeated by a given node – must be changed to move information more than 30 km to other sector nodes (see the September 1978 SATS, Sec. II-E).

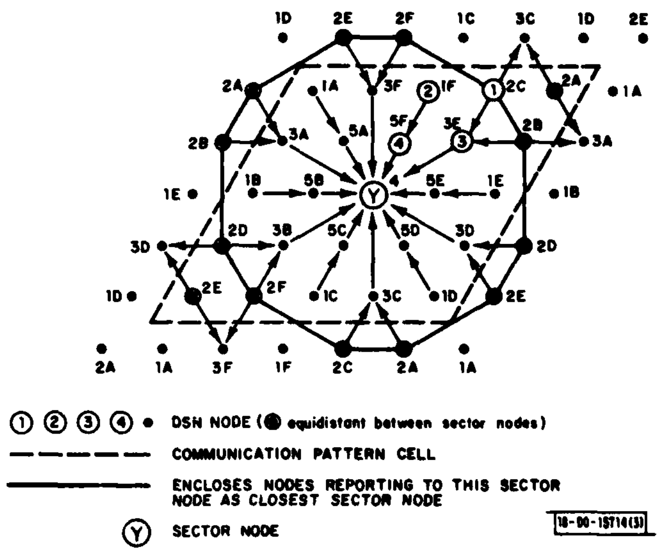


Fig. I-4. Report circuit communication pattern. Y is a sector node and nodes numbered 1, 2, 3, and 4 are typical nodes which must distribute reports to the sector node.

Figure I-4 gives a particular pattern of node broadcasts which was proposed in the September 1978 SATS. The broadcast order of the nodes there is 1A, 1B, 1C, 1D, 1E, 1F, 2A, 2B, . . . , 3F, 4, 5A, 5B, 5C, 5D, 5E, 5F, and then repeats. All nodes with the same pattern label broadcast simultaneously. In the example, all simultaneously broadcasting nodes are 50 km apart. The primary consideration in keeping simultaneously transmitting nodes far enough apart is that no receiving node should have to listen to two transmitters at once. (Also see the September 1978 SATS, Sec. II-F-3.) This particular pattern was selected because it was felt that careful ordering of transmission sequence would speed the transfer of information to the sector nodes. One of the results of the analysis reported here is that such a carefully ordered circuit is not faster on the average than a randomly ordered circuit.

Finally, when a node receives a report packet, the information contained therein cannot be rebroadcast until the node has had time to process the packet. For the following analysis we assume that this processing takes up to 0.25 sec. Also, we allow nodes to rebroadcast critical report information up to 6 times to obtain reliable transmission.

TABLE I-5

PROBABILITY OF REPORT CONNECTION AND COMMUNICATION DEFECTS FOR NODE 1†

RETRANSMISSION DELAY OF 7 TRANSMISSION TIME SLOTS
EACH NODE RETRANSMITTING EACH DATUM 6 TIMES

A	B	C	Probability of Report Communication Defect for S =									D
			2	3	4	5	7	9	11	13	15	
100	0.0	0.9	0.49	0.06	0.02	0.01	0.00	0.00	0.00	0.00	0.00	4
		0.99	0.94	0.34	0.07	0.04	0.01	0.00	0.00	0.00	0.00	6
90	0.02	0.9	0.57	0.10	0.02	0.01	0.00	0.00	0.00	0.00	0.00	4
		0.99	0.98	0.42	0.10	0.04	0.01	0.00	0.00	0.00	0.00	6
80	0.02	0.9	0.68	0.24	0.07	0.03	0.01	0.00	0.00	0.00	0.00	6
		0.99	0.97	0.58	0.27	0.14	0.03	0.01	0.00	0.00	0.00	9
70	0.05	0.9	0.80	0.35	0.15	0.08	0.02	0.01	0.00	0.00	0.00	7
		0.99	1.00	0.73	0.38	0.25	0.08	0.03	0.01	0.00	0.00	10
60	0.15	0.9	0.89	0.42	0.22	0.09	0.02	0.01	0.01	0.00	0.00	7
		0.99	1.00	0.87	0.53	0.33	0.10	0.03	0.01	0.01	0.00	10
50	0.23	0.9	0.88	0.50	0.27	0.09	0.03	0.01	0.00	0.00	0.00	8
		0.99	1.00	0.89	0.64	0.41	0.12	0.04	0.02	0.01	0.00	10
40	0.40	0.9	0.94	0.56	0.34	0.15	0.03	0.01	0.00	0.00	0.00	9
		0.99	1.00	0.95	0.74	0.51	0.20	0.06	0.02	0.00	0.00	11
30	0.58	0.9	0.98	0.62	0.39	0.18	0.04	0.02	0.00	0.00	0.00	8
		0.99	1.00	0.99	0.86	0.61	0.24	0.06	0.03	0.01	0.01	12
20	0.72	0.9	0.98	0.63	0.41	0.12	0.04	0.01	0.00	0.00	0.00	8
		0.99	1.00	0.98	0.92	0.75	0.25	0.09	0.02	0.01	0.00	11
10	0.91	0.9	1.00	0.62	0.31	0.09	0.00	0.00	0.00	0.00	0.00	7
		0.99	1.00	1.00	0.98	0.70	0.25	0.06	0.00	0.00	0.00	10

A Percentage of Nodes Functional
B Probability of Report Connection Defect
C Required Probability of Message Arriving after S sec
D Smallest S with Probability of Report Communication Defect Less Than or Equal to 0.02

† Calculated by Monte Carlo simulations of 400 situations each with 1000 data items.

1. Report Circuit Defects

Two kinds of defects are possible in the report circuit: these are connection defects and communication defects. Connection defects correspond to the situation where no possible communication path exists from a source node to the sector node. A communication defect is when such a path exists but the strawman communication algorithms fail to achieve the desired transmission. These are described in more detail below and their incidence quantified as a function of percent functional nodes in the DSN.

Communication defects are defined in terms of probability. The definition specifies that a defect occurs when report messages from node X get to the nearest sector node Y within S sec with probability less than some specified P. For example, we may be interested in a defect in which reports from X get to Y within 3 sec with probability less than 0.99. The defect probability is a function of the number of seconds S allowed for transmission, the probability of success P required for there to be no defect, the proportion of nodes that are not functional, the relative geometry of X and Y in the DSN, and the communication pattern. In what follows, the destination Y is the nearest sector node to the source node X. Both of those nodes are assumed to be functional for the analysis.

Because of hexagonal and reflection symmetries, there are only 4 types of node positions to be considered. These are shown in Fig. I-4 as nodes 1 through 4. Of course such a regular grid is not essential for the DSN but is used for convenience.

The definition of a connection defect is less subtle than for a communication defect. It is the situation where there is no route at all to the sector node. However, not all global routes are allowed in this definition. Each node has its own 30-km area of interest. As a result, very indirect or circuitous routes may never be used even if they exist. Changes of the size and shape of the areas of interest would influence connection defects to some extent. In what follows, communication defects are considered only for cases without connection defects.

Table I-5 lists the probabilities of report connection and communication defects for X = node 1, for P = 0.9 or 0.99, for S in the range from 1 through 15, for various probabilities of an arbitrary node being functional, and for the communications pattern of Fig. I-4. The first column is the percentage of functional nodes, or the probability of an arbitrary node in the subnetwork (other than X or Y) being functional. The second column is the probability of encountering a report connection defect. The third column is the required probability P of the datum arriving after S sec. The next 9 columns are report communications defect probabilities for various values of S. The last column is the smallest S value such that data arrive after S sec with probability P in 98 percent of all situations; in other words, the smallest S with report communication defect probability less than or equal to 0.02. This last column measures the delay in reliably getting a datum from X to Y.

As an example, consider node 1, the farthest node from the sector node, in a DSN with 70 percent of all nodes functional. Then the probability of node 1 and the sector node being disconnected is 0.05. If node 1 is connected to the sector node, the probability of being in a situation where less than 99 percent of all data from node 1 arrive at the sector node within 2 sec is 1.00, within 3 sec is 0.73, within 5 sec is 0.25, and within 9 sec is 0.03. The number of seconds that must be allowed for 99 percent of all data to arrive in 98 percent of all situations is in the last column, and is 10 sec.

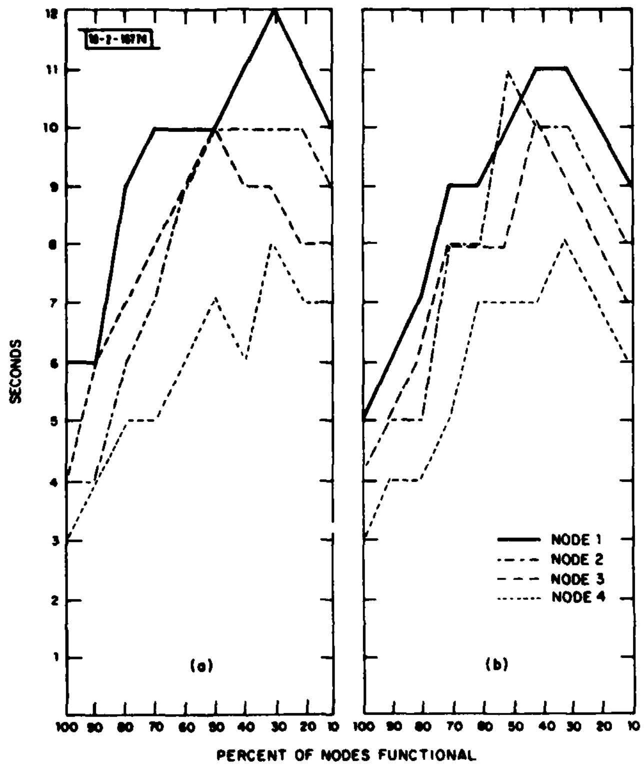


Fig. I-5. Strawman report circuit transport delay from source node to sector node. Delay plotted against the percent of functioning nodes in the DSN. Delay shown is smallest delay with probability of communication defect less than or equal to 0.02 for a required probability of arrival equal to 0.99. Results shown for typical nodes 1, 2, 3, and 4 of Fig. I-4. Processing delay within each node set at 0.25 sec. (a) For a patterned circuit designed to move reports quickly to the sector node and (b) for a random circuit only constrained to avoid packet collisions.

Monte Carlo analysis methods were used to compute this table. The procedure was as follows. First a DSN situation was created using a random number generator to decide which nodes and links were functional, based on the given probabilities for nodes and links being functional. If the originating node and sector node were not disconnected, then 1000 data items were sent from the originating node 1, 2, 3, or 4 to the sector node. A histogram of the number of seconds from initial broadcast until the datum arrives at the sector node was constructed. This histogram calculation was repeated for 4000 situations, 400 for each of 10 probabilities of node functionality (100 percent through 10 percent as in the tables). From these results the probabilities in the tables were estimated.

Some important additional details of the simulation of the multihop transmission process are as follows. In transmitting the data a random number generator was used to determine whether a transmitted packet was received by a given node with a usable link to the transmitter. Each second of time was represented by a sequence of 25 transmission slots. To account for processing delay between reception of information by a node and allowable retransmission of that information by the node, the datum was not retransmitted by any node that had not received it at least N report circuit slots earlier. For Table I-5, which is for a nominal 0.25-sec delay, N = 7 was used. Also in the case of Table I-5, once the datum was received by a node, it was rebroadcast exactly 6 times, after which it was not broadcast again. This is the strawman scheme for critical data and if data are not critical, they may be replaced by a more recent datum before being rebroadcast 6 times, but the more recent datum should be equivalent to the original datum for our current purposes.

The last column of Table I-5 can be plotted as a measure of the delay vs the percentage of functional nodes. Figure I-5(a) contains such a graph for node 1 from Table I-5, and similarly constructed graphs for the other nodes 2, 3, and 4.

Figure I-5(b) was made in the same way as Fig. I-5(a) except that the circuit pattern is randomly ordered for each of the 4000 situations. No attempt was made to establish a pattern to move data to and from sector nodes faster than a random pattern would. Figure I-5(b) is also valid if the sector node is replaced by a randomly chosen node from which information about targets within 30 km is to be extracted.

Comparison of Figs. I-5(a) and (b) shows that the ordered circuit pattern is no better than a completely random circuit pattern. It would appear that we can give up trying to provide a carefully ordered pattern for the report circuit, and use randomly established circuit patterns instead.

Figure I-6(a) shows the effect of changing the processing delay between the time a datum is received and the time it can be rebroadcast. Delays of approximately 0.25, 0.1, and 0 sec are represented. Changing this delay does not have much effect. Figure I-6(b) can be compared with Fig. I-6(a) to show that the ordered circuit pattern is no better than a completely random pattern even with shorter delays between reception and retransmission.

Figure I-7 shows the difference between requiring messages to arrive in a certain time with 0.9 probability, and the same requirement with 0.99 probability. The higher probability requirement lengthens the transmission time by perhaps 20 to 30 percent.

Table I-6 is the same as Table I-5 but the number of times a datum is retransmitted by a node has been changed from 6 to 4. Except for statistical fluctuations consistent with the number of random cases studied, these two tables are the same except for the case of P = 0.99 and large S. For example, the 50-percent functionality communication defect probability is 0.29 for

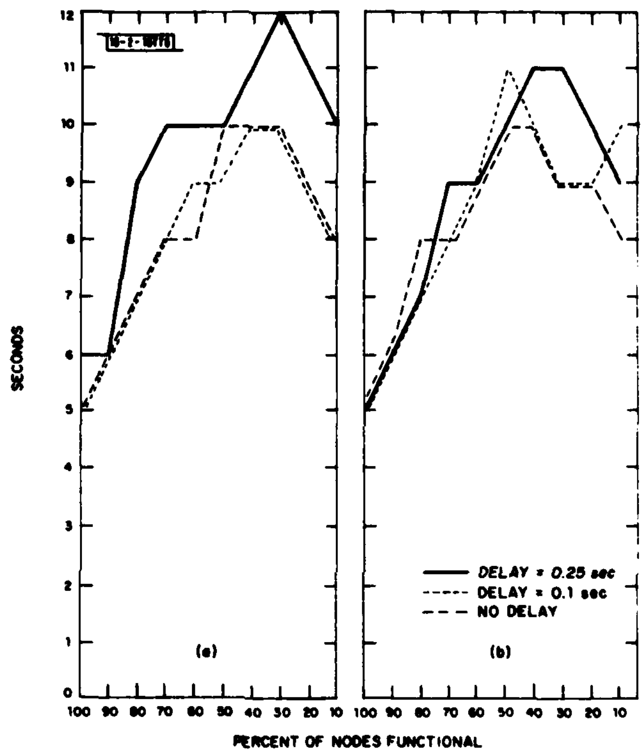


Fig. I-6. Transport delay as a function of percent functional nodes as in Fig. I-5 except that delays are shown for only node type 1 but for single-node processing delays of 0.0, 0.125, and 0.25 sec. (a) For a patterned circuit and (b) for a random circuit.

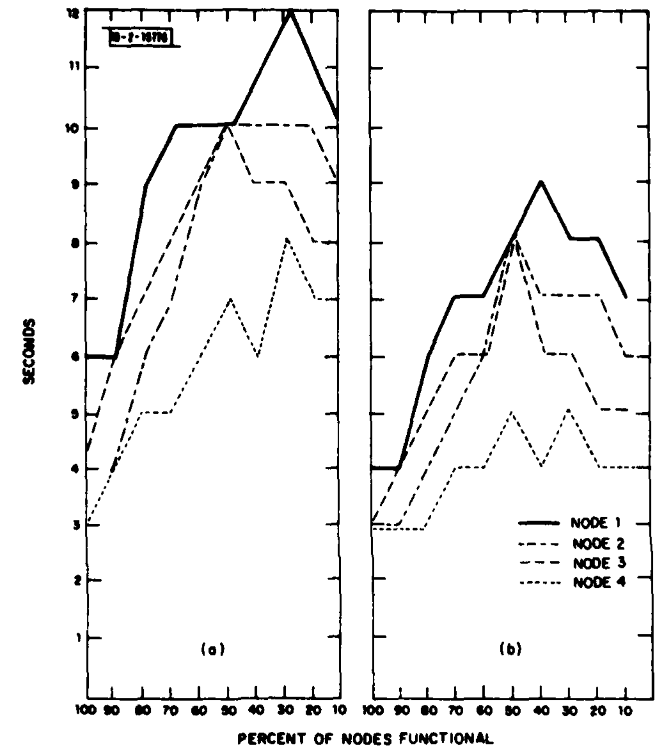


Fig. I-7. Transport delay as a function of percent functional nodes. Figure (a) is the same as Fig. I-5(a); Figure (b) is the same with the required probability of arrival reduced to 0.9 from 0.99.

TABLE I-6

PROBABILITY OF REPORT CONNECTION AND COMMUNICATION DEFECTS FOR NODE 1†

RETRANSMISSION DELAY OF 7 TRANSMISSION TIME SLOTS
EACH NODE RETRANSMITTING EACH DATUM 4 TIMES

			Probability of Report Communication Defect for S =									
A	B	C	2	3	4	5	7	9	11	13	15	D
100	0.0	0.9	0.51	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	4
		0.99	0.95	0.29	0.05	0.03	0.00	0.00	0.00	0.00	0.00	6
90	0.0	0.9	0.56	0.12	0.04	0.02	0.00	0.00	0.00	0.00	0.00	5
		0.99	0.97	0.40	0.14	0.07	0.04	0.04	0.04	0.04	0.04	
80	0.03	0.9	0.71	0.25	0.07	0.03	0.01	0.01	0.00	0.00	0.00	6
		0.99	0.99	0.61	0.26	0.12	0.05	0.04	0.04	0.04	0.04	
70	0.06	0.9	0.77	0.36	0.15	0.05	0.00	0.00	0.00	0.00	0.00	6
		0.99	0.99	0.69	0.40	0.26	0.11	0.10	0.10	0.10	0.10	
60	0.11	0.9	0.84	0.38	0.18	0.06	0.02	0.00	0.00	0.00	0.00	6
		0.99	1.00	0.81	0.46	0.30	0.14	0.12	0.12	0.12	0.12	
50	0.23	0.9	0.90	0.56	0.33	0.17	0.04	0.01	0.01	0.00	0.00	8
		0.99	1.00	0.90	0.71	0.51	0.34	0.30	0.29	0.29	0.29	
40	0.41	0.9	0.91	0.56	0.31	0.11	0.03	0.01	0.00	0.00	0.00	8
		0.99	1.00	0.92	0.72	0.57	0.36	0.34	0.34	0.34	0.34	
30	0.59	0.9	0.94	0.59	0.41	0.11	0.03	0.00	0.00	0.00	0.00	8
		0.99	1.00	0.94	0.83	0.65	0.50	0.49	0.48	0.48	0.48	
20	0.76	0.9	0.94	0.54	0.32	0.09	0.03	0.00	0.00	0.00	0.00	8
		0.99	1.00	0.98	0.88	0.76	0.66	0.63	0.63	0.63	0.63	
10	0.89	0.9	1.00	0.44	0.24	0.03	0.00	0.00	0.00	0.00	0.00	6
		0.99	1.00	1.00	0.98	0.90	0.79	0.79	0.79	0.79	0.79	

A Percentage of Nodes Functional

B Probability of Report Connection Defect

C Required Probability of Message Arriving after S sec

D Smallest S with Probability of Report Communication Defect Less Than or Equal to 0.02

† Calculated by Monte Carlo simulation of 400 situations each with 1000 data items.

large values of S with P = 0.99. What this means is that in 29 percent of the connected situations, data are completely lost more than 1 − 0.99 = 1 percent of the time. Such loss occurs because a node stops transmitting a datum before the next nodes in a path have received it. The report circuit as defined here would require that each critical datum be rebroadcast at least 6 times to ensure reliable delivery of 99 percent of all critical data.

It should be noted that the above results ignore the fact that as node functionality decreases, the network could compensate by having each remaining node broadcast more frequently. For example, with 50-percent node functionality, each node might broadcast twice as often, and the transmission delays might be half of what our analysis indicated. Such adaptation was not part of the original strawman scheme but could easily be incorporated.

Finally, Fig. I-8 shows disconnection probabilities taken from Table I-5 for node 1 and from other Monte Carlo run results for nodes 2, 3, and 4. The disconnections are considerable only when less than 50 percent of the nodes are functional.

2. Report Circuit Packet and Message Formats

A report circuit packet consists of tracking report messages of various types and control messages. As with the sensor circuit, packet and message formats for the strawman report circuit (excluding control messages) have been further specified. This refinement is a step toward actual software which might be run on a simulated or experimental system. The formats are closely related to detection and tracking activity. As such they apply to our specific approach to tracking. Other approaches would call for different packet and message formats.

The main idea behind the very preliminary report message formats which have been defined is to report target locations and associated segments of possible position curves (see the March 1978 SATS,† Sec. III-B) which support the location. The first version has not included data quality or reliability measures in the message although these can and eventually must be added. As a report passes through the DSN each node in the reporting circuit can modify locations and data associations and can even discard tracking messages and create new ones for completely new targets. Obviously each node, including the user node, has to perform complicated track initiation and tracking functions to participate in the DSN.

Each of the strawman tracking report messages which has been designed starts with a header containing message type identifier, a time and location for the target, and perhaps a few past time-location values for the same target. In addition, the header identifies the node from which the report was obtained and provides the target identifier assigned by that node for this target.

After the tracking report header there are any number of sensor tracking report submessages. For acoustic measurements these give the identifier of the sensor node making the measurements, the identifier assigned the target by that node, the end points of a measurement derived possible position line segment on which the target is presumed to be located (see the March 1978 SATS, Sec. III-B), and a target-type estimate based on the acoustic spectrum heard by the node's sensor. If the node from which the tracking report message was received is also one of the nodes furnishing a submessage, then the target identifier in that node submessage is the same as the identifier in the header. The submessage structure for radar has not been defined.

† Distributed Sensor Networks SATS, Lincoln Laboratory, M.I.T. (31 March 1978), DDC AD-A060685.

As currently defined, each strawman tracking message consists of less than 100 bits of header plus about 50 bits per acoustic sensor submessage. A tracking report message with 3 acoustic submessages would be less than 250 bits long. Five such messages could be included in a 1500-bit report circuit packet with 250 bits left over for control information. To handle 10 targets as is required for the strawman system would distribute the reports over two report circuit packets over a 2-sec interval.

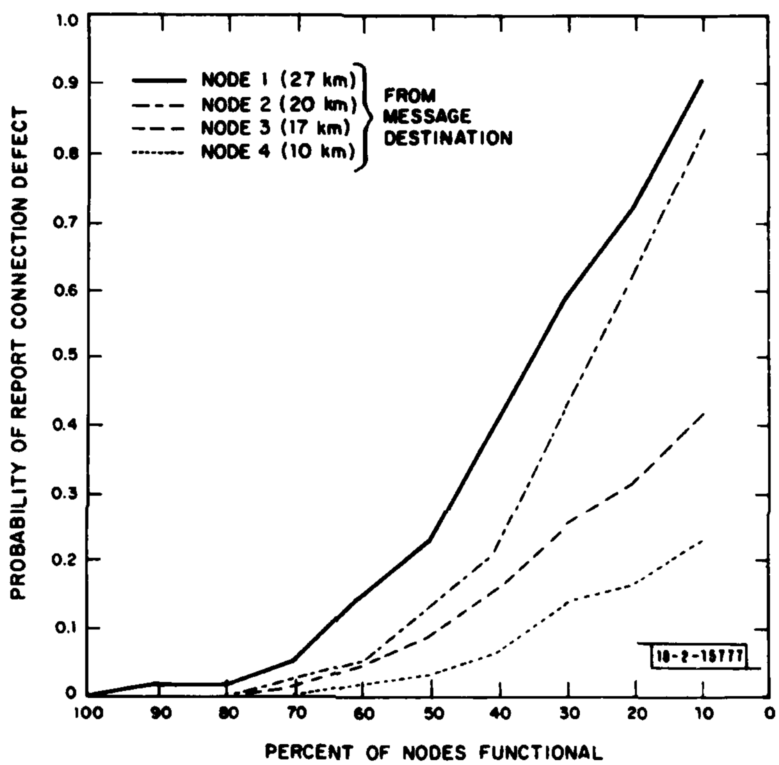


Fig. I-8. Probability of report connection defect for typical nodes 1, 2, 3, and 4 of Fig. I-4. Probability shown as a function of percent nodes functional in the strawman DSN.

II. SYSTEM STUDIES

Several system studies have been completed and are reported in this section. These include a quantitative study to determine the importance of redundant DSN sensor coverage, a study to determine the relative performance of adaptive and nonadaptive communications in a DSN, a study of cost and system utility issues for a nonuniform DSN deployment, and a preliminary design study for a DSN approach to a weapons location system.

A. REDUNDANCY EFFECTIVENESS IN A DSN

A study of sensor redundancy in a DSN has been completed. The purpose of this study was to determine how DSNs might degrade in effectiveness if nodes become inoperative due to natural failures or failures resulting from enemy action. It is intuitively obvious that the more nodes and the more overlap we have in a given area, then the greater will be the redundancy. The objective of this study was to quantify the effectiveness of this redundancy.

To minimize complication, the study was limited to sensors which can individually locate targets and for which the area of effectiveness is a circle located at the sensor. A normal monostatic radar, even one with poor range or azimuth resolution, can be modeled in this way although the coverage area of a single radar for low-flying aircraft is generally a complicated shape and not a circle. Extensions of the analysis to sensors for which the coverage area is more complicated or for which multiple sensors are required for location are possible but have not been completed.

For the study, the DSN nodes were located on a hexagonal grid as shown in Fig. II-1. Each DSN node was assigned the same effective detection radius as shown by the circles around the nodes in Fig. II-1. Finally, a box was drawn around the sensor range circles to represent the limits of system coverage.

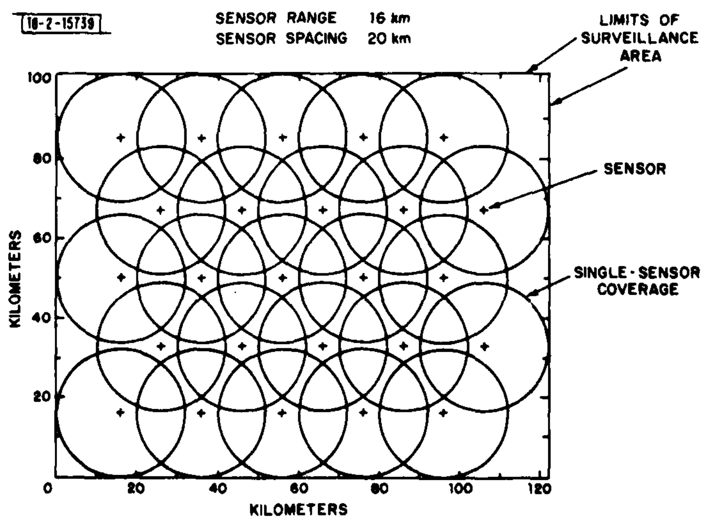


Fig. II-1. Location of nodes for 25-node DSN; centers shown by crosses, coverage shown by circles.

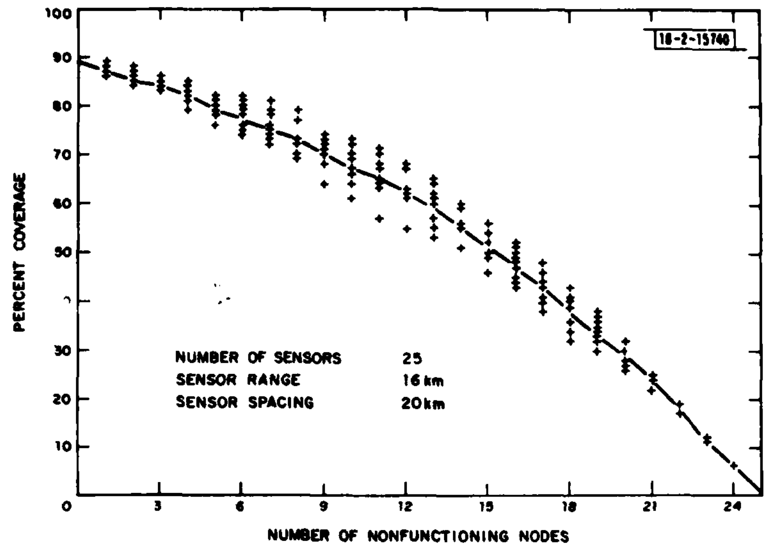


Fig. II-2. Reduction in percentage cover as nodes are eliminated for 25-node DSN; 10-trial Monte Carlo simulation. Solid line is average result and crosses indicate individual run results.

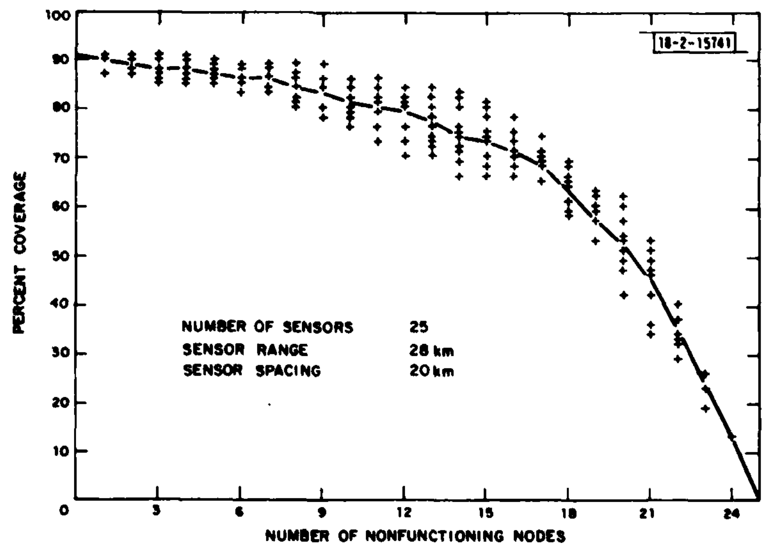


Fig. II-3. Percentage cover as a function of sensors eliminated for 25-node DSN with sensor range of 28 km.

As a measure of system effectiveness, we chose the percent of the area covered by sensors within the box representing the limits of coverage. This was chosen as it directly related effectiveness to the coverage of a rectangular area. It has the advantage that the area coverage by a different number of sensors can be readily compared. However, its disadvantage is that even with all sensors effective the coverage is less than 100 percent.

For this study, the user of the system (say a command post) is assumed to have direct communication access with all nodes in the system. This is assumed to be the case for all active nodes in the system no matter how many other nodes have been eliminated. The related study reported in Sec. B addresses the effect of adaptive and nonadaptive communication schemes on the results reported here.

The basic approach has been to assign sensor separations and ranges and then to deploy the sensors on an appropriate regular grid and bound the sensor coverage area with a box as mentioned above. Then a random number generator was used to select nodes at random to be removed from the system. The percent area covered was calculated and plotted as a function of the cumulative number of nodes eliminated. Figure II-2 shows the results obtained from 10 trials for the 25-node DSN shown in Fig. II-1. Individual results are indicated by crosses and the average result is indicated by the solid curve.

Note that the average curve, and this is also true of the individual runs, does not decrease linearly with the number of nonfunctioning sensors. This is a result of the amount of sensor redundancy in the system at the start. For no redundancy at all, for example, if sensor range were one-half of the separation between sensors, the curve would be a straight line. However, when the sensor range is large compared to separation the redundancy at the start is large and a substantial fraction of the nodes can be removed without reducing system coverage. This is demonstrated by the results shown in Fig. II-3. It is the result of another series of 10 runs with the same sensor spacing but with the sensor range increased to 28 km from 16.

Because of sensor, propagation, and terrain factors, as has been previously noted for the strawman design presented in our previous SATS, sensor separations of 20 km would be very large for any DSN designed for cruise missile or low-flying aircraft surveillance and tracking. Figures II-4 and -5 show the results from 10 trials for sensor separations of 10 km, a total of 100 sensor sites, and sensor ranges of 7 and 14 km. Such numbers are more appropriate for the strawman system. The total area of the surveillance box remained roughly 100 km squared.

Note that the trends of Figs. II-2 and -4 are quite similar as are those of Figs. II-3 and -5. It is not the number of sensors, but the ratio of the sensor range to the sensor separation which is the important parameter. Of course this is so because the issue is redundancy. If the area covered were large enough to make edge effects vanishingly small, and the number of trials were large enough, the curves showing average effective coverage should be identical for similar separation to range ratios if it is plotted against percent nonfunctioning nodes rather than the absolute number of nodes.

These observations led to the selection of another comparison parameter. This is the average percent reduction in percent coverage when half the original nodes have been lost due to hostile action or maintenance problems. Figure II-6 shows percent reduction plotted as a function of the ratio of effective range to spacing for 25, 49, and 100 nodes. This demonstrates in a slightly different way that the number of nodes is not a prime factor but that the range to separation ratio is the important factor.

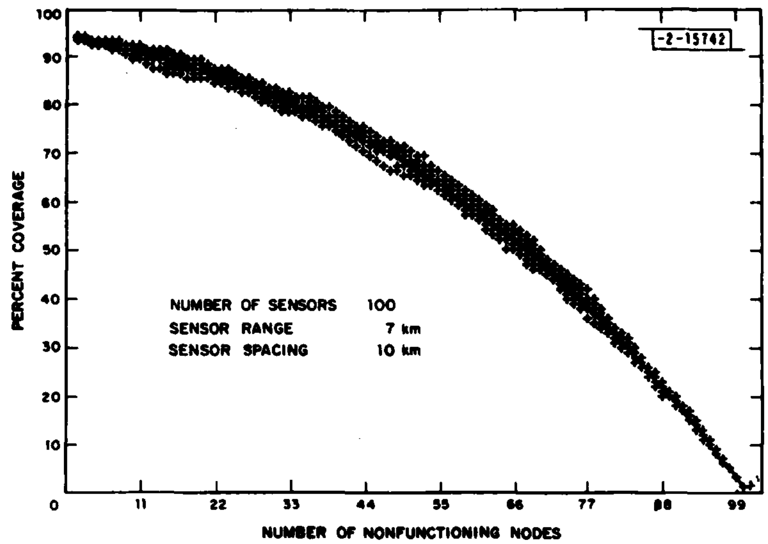


Fig. II-4. Percentage cover as a function of sensors eliminated for 100-node DSN with a sensor range of 7 km.

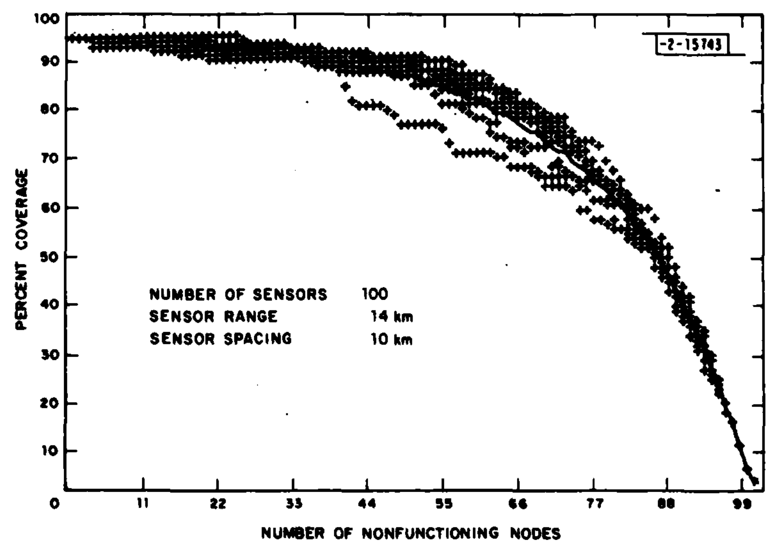


Fig. II-5. Percentage cover as a function of sensors eliminated for 100-node DSN with a sensor range of 14 km.

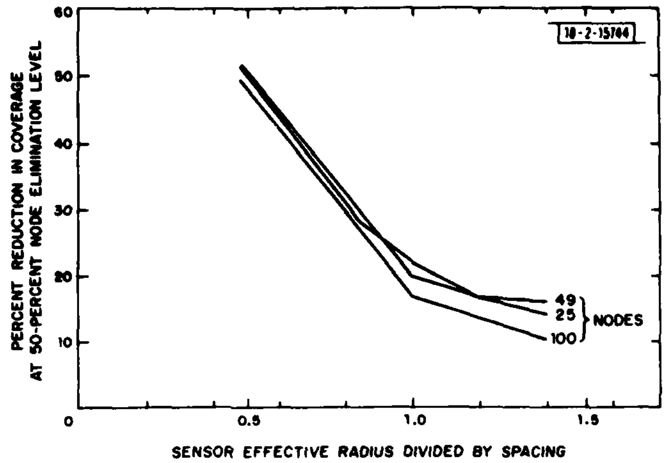


Fig. II-6. Percent reduction in coverage at 50-percent node elimination level as a function of the ratio of effective sensor radius to sensor spacing. Approximate area covered was 100 km square. Curves shown for 25-, 49-, and 100-node DSN.

For the measure of effectiveness shown in Fig. II-6, there is clearly a point of diminishing returns in increasing the sensor range relative to the sensor separation. The knee in the curves occurs when range and separation are about equal. At that point there is only a 20-percent reduction in coverage for a 50-percent loss of nodes. It is clear that a very large increase in sensor range would be needed to reduce this to only 10 percent.

Similar results would be obtained for the more complicated case when multiple sensor detections would be required to obtain useful target positions. We anticipate that slightly larger detection range to separation ratios might be desired in the multiple sensor case.

B. ADAPTIVE COMMUNICATIONS IN A DSN

A study has been completed of the relative performance of adaptive and nonadaptive communications in a DSN. As in Sec. A above, the performance was measured by computing the percent area covered by sensors as a function of the number of nodes eliminated. However, the utility of a sensor node depends not only upon its own existence but upon the existence of other nodes which are used to relay information from the sensor to a command center making use of the DSN. Thus if information from node i passes through node j in reaching the command center and node j is eliminated then, effectively, so is node i. This is the nonadaptive situation. The adaptive situation would be when the system adapts to find an alternative to the node j route if node j is eliminated. If such a route exists and is found, then node i can continue to contribute to the system.

It was determined that adaptive communications is significantly superior to nonadaptive techniques. In the 25-node case studied, the adaptive system closely approximated the performance of a system with perfect communications up to a level of 30-percent node loss. This was for a system with the minimum single-hop communication ranges which would actually allow any communication or adaptation. For larger single-hop ranges, the adaptive system can approximate the ideal case for even greater node loss. Of course, in the limit, if the single-hop range

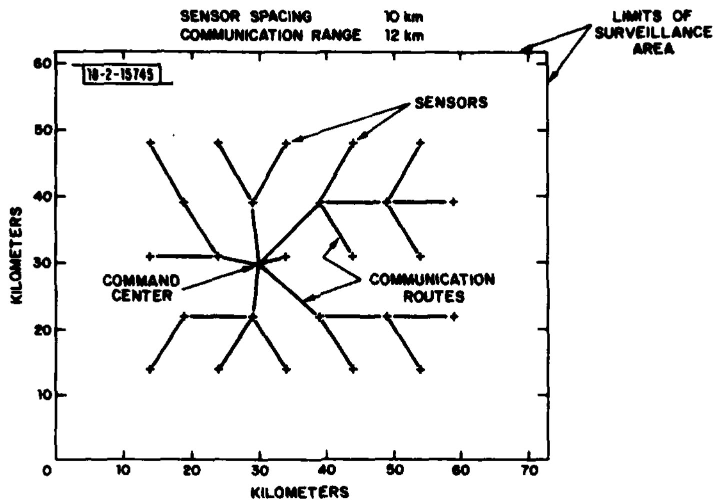


Fig. II-7. Communication links for 25-node network as selected by rings of power algorithm. Nodes located on a 10-km grid and single-hop communication range equal to 12 km. Crosses are the node locations and the box is the area of surveillance when sensor ranges are set to 14 km.

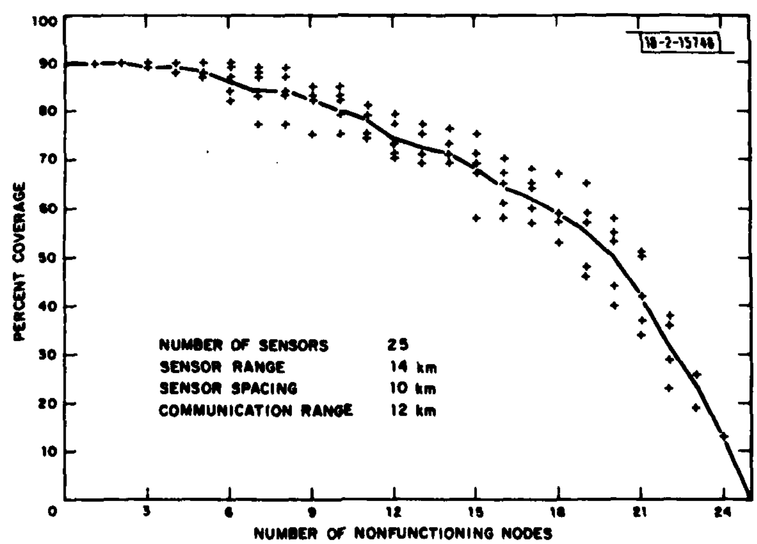


Fig. II-8. Monte Carlo simulation of degradation of DSN performance as nodes are eliminated with perfect communications.

is very large the system becomes the perfect communication case since each node can communicate directly with the command center.

We start by positioning the nodes. In this study 25 nodes were placed on a 10-km hexagonal grid as shown in Fig. II-7 which also shows communication routes to a command center which will be discussed shortly. The sensor range, which is not shown on the figure, was set to 14 km. The box representing the area to be covered was constructed in the same way as the boxes in Sec. A. It is the smallest rectangular box containing all the area covered by any of the sensors. Then a command center position is chosen. This command center is not a node, it is simply the recipient of all the messages.

In order to perform this study we had to devise an algorithm for adaptively organizing the communications links. The one we chose has been called the "Rings of Power" algorithm as it approximates the sociological phenomena surrounding a head of state. The command center is the center of power and is assigned a level of 0. Any nodes that can communicate directly with the command center are assigned a level 1. Then other nodes are connected by extending the ring of power outward. The selection of links is made by a node choosing to link to the lowest level node he can communicate with, and if there is equal choice, the node will choose the shortest distance. If more than one option still exists then one is arbitrarily selected. The node then gets assigned one level higher than the node to which he is connected. It is this algorithm, with a 12-km communications range, which derived the communications links shown in Fig. II-7. The 12-km communications range simply means that a node can communicate with every node which is within 12 km of itself. No more sophisticated modeling of link usability or performance was used in this study.

Monte Carlo simulations were run for three different situations to determine the effect of node elimination. First we assumed that each of the nodes had perfect communications with the command center independent of the links chosen by the algorithm. This was the only situation considered in Sec. A and is the baseline for comparison. Running 5 trials gave the results shown in Fig. II-8. In this figure the results of the individual trials are marked by crosses and the average result by the solid line. We next ran a nonadaptive communications simulation using the communications links of Fig. II-7. In this, we assumed that the elimination of a node also eliminated coverage by nodes that communicated through it. The results of a set of 5 trials are shown in Fig. II-9. The final test was to allow the network to adapt its communication links using the rings of power algorithm as nodes were eliminated. The results for this case are shown in Fig. II-10.

From these results it can be seen that the adaptive scheme, initially, closely approximates the perfect communications case. Only when more than 50 percent of the nodes have been eliminated does the performance significantly degrade over the ideal and it is equal to the ideal for up to about 30-percent node loss. On the other hand, using nonadaptive communications results in a rapid degradation in performance as nodes are eliminated. There is considerable scatter in the data. This probably results from a rapid degradation in network performance when certain key nodes are eliminated.

Similar runs were done with the communications distance increased to 18 km. This allowed each node to communicate with the ring of second-nearest neighbors as well as with the ring of nearest neighbors. Communication connectivity was increased in the network. This resulted in significant changes in communication patterns. The initial routing to the command center, as set up by the rings of power algorithm, is shown in Fig. II-11. The results for the perfect, nonadaptive, and adaptive cases are shown in Figs. II-12, -13, and -14. With the higher level of

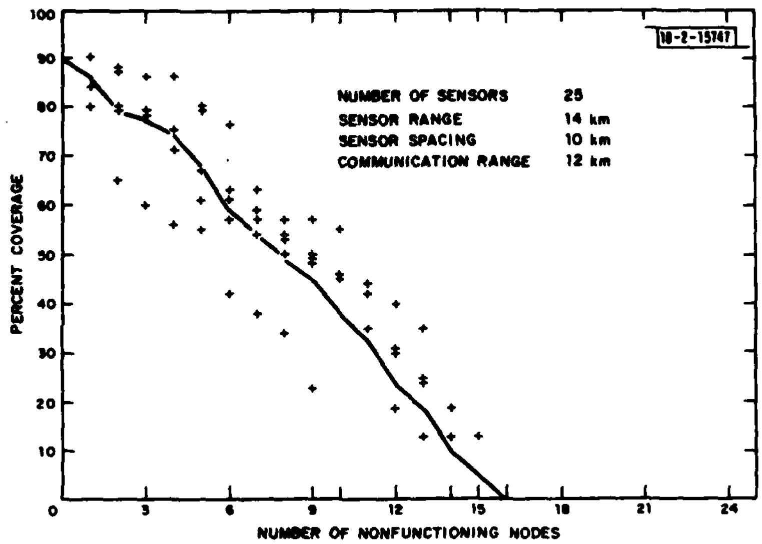


Fig. II-9. Degradation in effectiveness of DSN as nodes are eliminated with nonadaptive communications.

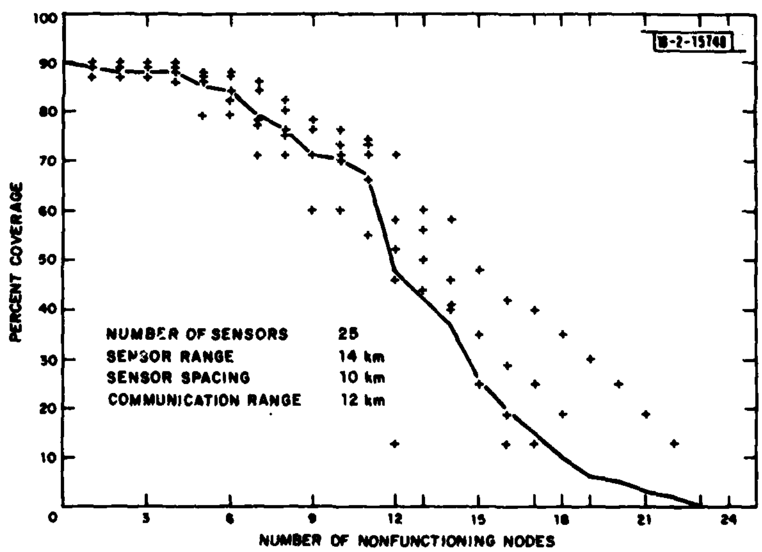


Fig. II-10. Degradation in effectiveness of DSN as nodes are eliminated with adaptive communications.

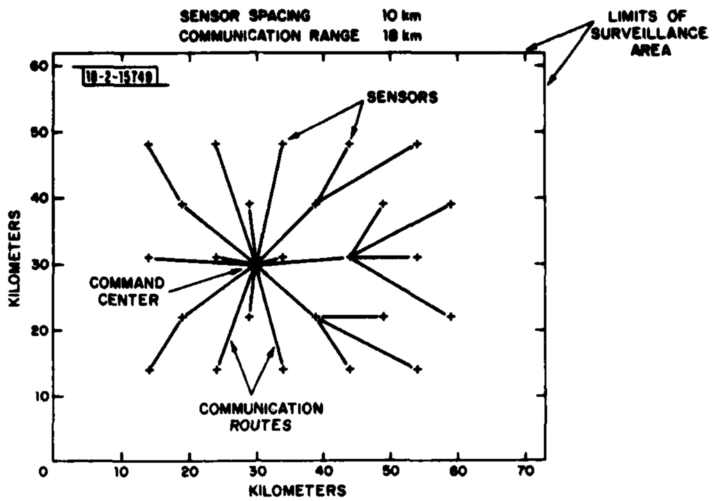


Fig. II-11. Communication routes to command center set up by the rings of power algorithm for 25-node network with 18-km single-hop range.

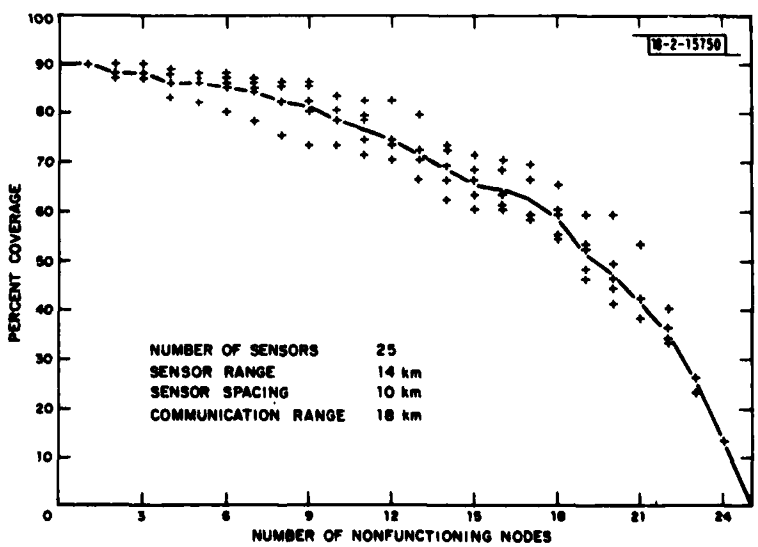


Fig. II-12. Degradation in effectiveness of DSN as nodes are eliminated with perfect communications to the command center.

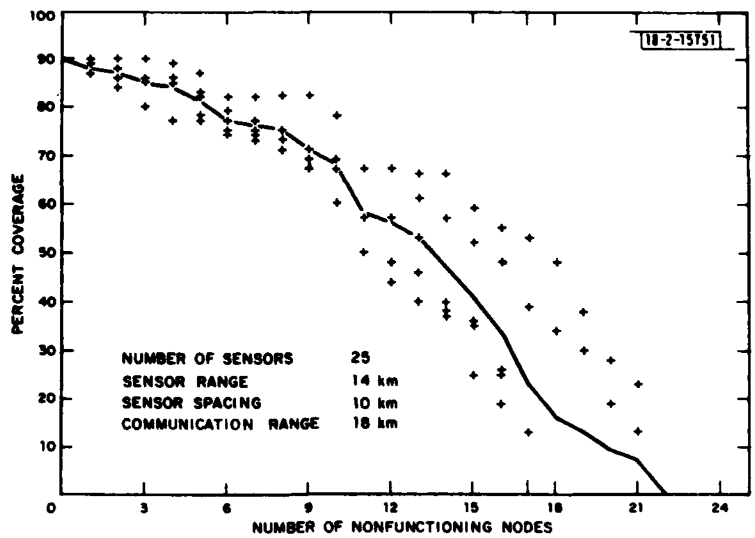


Fig. II-13. Degradation in effectiveness of DSN as nodes are eliminated with nonadaptive communications.

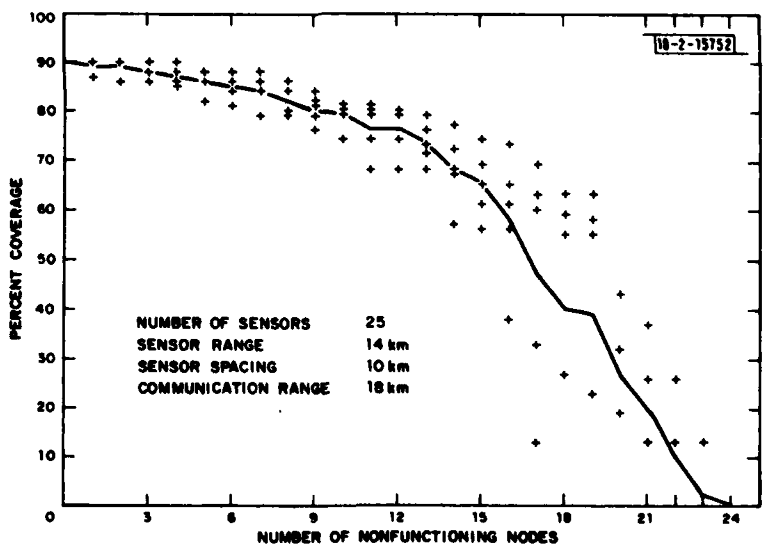


Fig. II-14. Degradation in effectiveness of DSN as nodes are eliminated in the case of adaptive communication routing set up by the rings of power algorithm.

communication overlap the adaptive algorithm more closely approximates the ideal communications case. Also, adaptive communications is still clearly superior to nonadaptive networks. Note that Fig. II-8 and -12 differ only statistically due to differences in the random sequence in which nodes were eliminated.

From these results, we can conclude that it appears preferable to use adaptive communications over straight nonadaptive networks. Also, much better redundancy performance is achieved if a node can communicate directly with second-nearest neighbors as well as nearest neighbors. However, it should be noted that larger single-hop ranges allow for more self interference and this can be a serious problem if the channel has limited capacity.

Note that the development of a real rings of power algorithm for the DSN has not been addressed here. No distributed implementation has been developed nor has the problem of multiple command centers been considered. Channel and link capability have not been taken into account. For example, the algorithms we studied could send most or all the traffic over a single overloaded link even when there are other alternatives. Our objective has been to demonstrate the importance of adaptation. To do that a plausible scheme was needed. The rings of power algorithm is such a plausible scheme.

Finally, we note that the adaptive reporting algorithm discussed here is quite different from the report circuits discussed in Sec. I. However, the report circuits do implicitly achieve adaptation by using communication redundancy. If a report has not been received from the node from which it is expected, any version received from another node is used. This is implicit rather than explicit adaptation but it does allow the strawman system to get the benefits of adaptive communication.

C. ANALYSIS OF NONUNIFORM DSN CONFIGURATIONS

In the 30 September 1978 SATS, in the context of a strawman system a nonuniform honeycomb sensor configuration was mentioned. In this section we review the cost advantage of that and other nonuniform geometries and indicate scenarios which make use of them. The discussion is in terms of very regular patterns of sensors only for convenience of exposition.

Figure II-15 shows a portion of a honeycomb deployment. As shown, the deployment has index k = 4 and depth 2. The index is essentially the number of steps along one side of the basic hexagonal honeycomb cell. The depth is the number of rows of sensor nodes in each cell wall. Both can be varied although here we treat only the index as a parameter. Note that for depth 2 we obtain a uniform filled grid of sensors for the case k = 1. In general, for depth d the honeycomb reduces to a uniform grid at k = d.

The average number of sensors per unit area decreases with increasing index. If the unit cost of a sensor node is held constant, this decrease in sensor density converts to a cost advantage relative to a uniform grid. Figure II-16 shows this cost advantage, which can be substantial. For example, for k = 5 it is about 2.75.

Of course, the cost savings of a honeycomb are valid only if the resulting system can still perform a useful service. The following simplified analysis demonstrates how a honeycomb surveillance system might be used in conjunction with manned interceptors to form an overall system which would be effective against low-level slow-flying targets.

Consider a large area covered by a honeycomb system and being penetrated by a number of targets of interest. Although the targets may change course we assume that sampling their positions as they pass cell walls is sufficient to obtain an overall timely picture of their location and

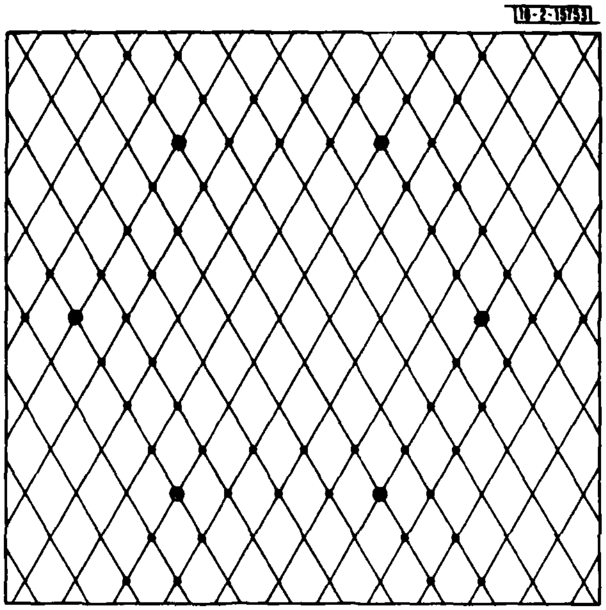


Fig. II-15. DSN honeycomb geometry with an index of k = 4 and a depth of 2.

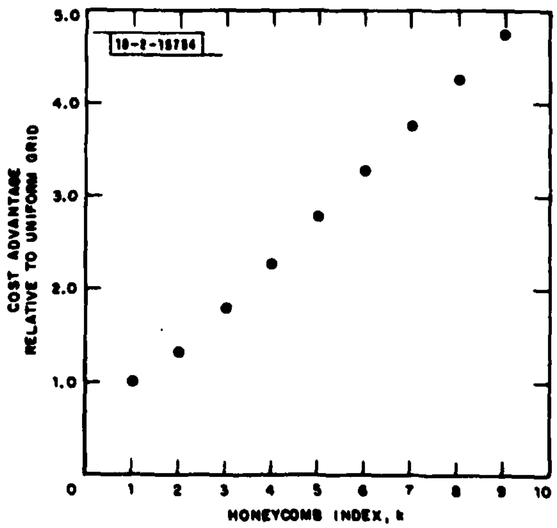


Fig. II-16. Cost advantage of a depth 2 honeycomb geometry relative to a uniform grid of DSN nodes.

ultimate destinations. We assume this is sufficient to deploy available interceptors to cells which contain targets. However, this coarsely sampled global information may not be precise enough to allow the interceptor to acquire the target.

Refer to Fig. II-17 for the following discussion of the intercept problem. The figure shows a honeycomb node configuration. The dotted lines define regions within which the system can track targets with sufficient accuracy to direct interceptors. As shown, the thickness of the

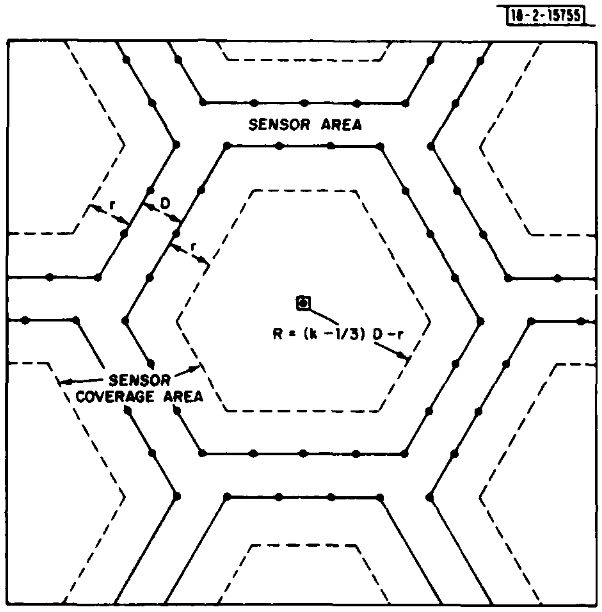


Fig. II-17. Sketch defining parameters used in analysis of DSN utility for directing interceptors. Sensor nodes given by dots; interceptor located at the center; dotted cell walls define the area in which targets will be directly sensed by sensors.

accurate tracking cell walls is $2r + D$ and it is shown as symmetric about the double line of sensors in each cell wall. Let $c = r/D$ and let $a = V/v$ where V is the interceptor velocity and v is the target velocity. Suppose the global system has positioned an interceptor at the square in the center of the cell and that its target is reacquired by the surveillance network as it moves away from the interceptor at the position indicated by the head of the arrow. The interceptor must close before the target enters the blind area in the next cell. It can then be shown that the interceptor will close in time if $a(1 + 2c)/(k + 5/3)$ is greater or equal to unity.

Figure II-18 shows the required speed advantage of the interceptor as a function of the depth 2 honeycomb index, k. It is shown for various values of c. For example, consider the case of k = 5 which would correspond to a cost advantage of 2.75. With c = 1, which we believe is a reasonable value, the interceptor needs a 2.2 speed advantage which is certainly practical for high-speed interceptors and subsonic targets of any kind. If c drops to 0.5 the required interceptor speed advantage rises to 3.3 which becomes more problematical.

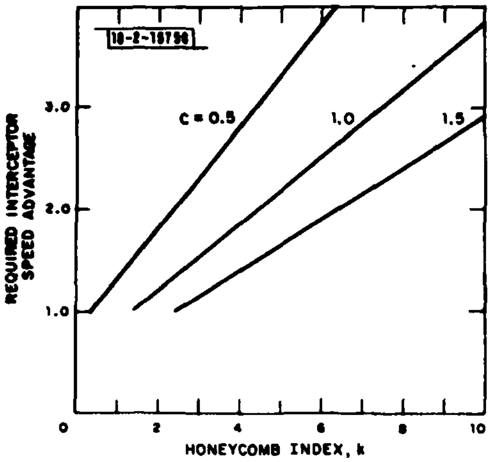


Fig. II-18. Required interceptor speed advantage over targets required for interception as a function of honeycomb index. The parameter c is a sensor coverage parameter c = r/D where r and D are defined in Fig. II-17.

The above is a very conservative analysis from the viewpoint of the DSN user. In many cases, the target will not take a fastest path through cell walls but will stay in the accurate track areas for substantially longer periods. Also, it assumes the point of penetration of a cell wall by a target is completely unpredictable. This will generally not be true. A target would need to trade in much of its range capability to make that even approximately true since it would require the target to make substantial random deviations from the direct route to its destination.

Other DSN configurations can provide cost advantages relative to uniform coverage and still provide a useful system. A linear barrier thick enough to allow vectoring of interceptors to targets could consist of multiple rows of sensors. The same considerations as above apply as long as targets penetrate the barrier without using the tactic of flying along it between rows of sensors. Such a tactic could not work if sensor locations were unknown to the targets. Even if they were it does not seem such a tactic would be very practical. Similar comments hold for a series of concentric sensor rings deployed about a small area as part of a terminal area defense system.

D. A DSN APPROACH TO HOSTILE WEAPON LOCATION

It is important to note that DSN technology does not represent a single fixed rigid system, but rather a design structure that is easily modified to satisfy new requirements. It is not restricted to specific sensors, targets, or scenarios. Once a DSN has been developed for one task it should be possible to develop new systems for other tasks with a short cycle time from problem definition to system development. To emphasize this adaptability we have formulated a DSN approach to tactical hostile weapon location. The overall design and system characteristics are given below.

Figure II-19 shows one possible deployment of a strawman DSN for locating firing hostile weapons. The system consists of 11 nodes deployed in two lines on the friendly side of the forward edge of the battle area (FEBA). Each node contains a packet radio, substantial amounts of computer power, a small (2- to 5-m aperture with 5 to 10 microphones) acoustic array, and/or a similar scale array of seismometers. This basic nodal hardware is roughly the same as that at an acoustic-only low-flying aircraft DSN node. Two of the nodes are augmented by projectile tracking radars for precision weapon location. The packet radios are assumed to have the capability of measuring range between nodes so that this information can be used to obtain relative node locations.

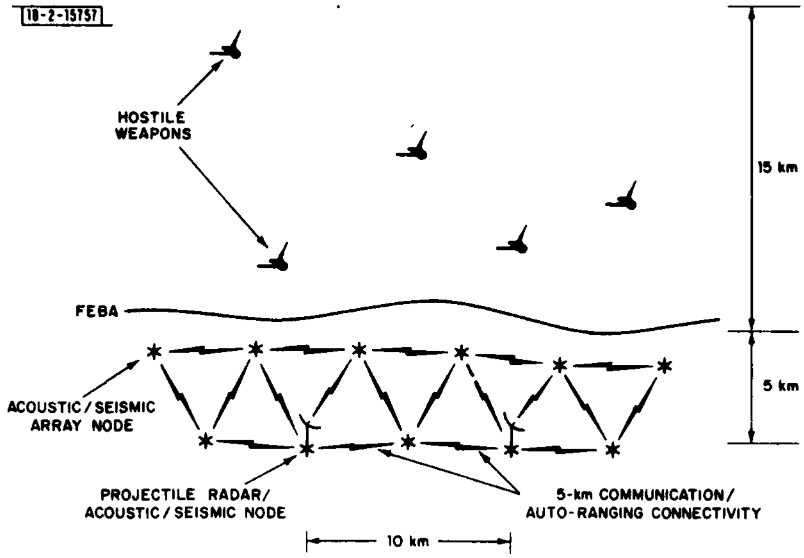
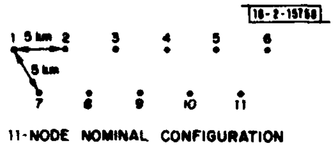


Fig. II-19. Deployment of a strawman 11-node weapon location DSN.

The DSN is highly interconnected by communication. This is to supply needed communication, to offer the opportunity to adjust to loss of nodes, and provide good self-location capability from internal node-to-node range measurements. Figure II-19 only shows the connectivity if radio range is 5 km. Larger ranges are required but the figure would be too complicated if all possible connections were shown, Figure II-20 shows the nominal connectivity for both 5- and 10-km radio ranges. Connectivity must be at least three to unambiguously locate a node without additional information. Since packet radios can easily be separated by 10 or more kilometers if there are no problems with multipathing and line of sight, the connectivity should be sufficient for self-location as well as communication. In practice, if approximate nodal location is known at deployment time, two-node connectivity may be sufficient to resolve gross position ambiguities which might otherwise require more connectivity.

The utility of acoustics for weapon location across the FEBA is well established. There is currently an acoustic system in military use. It is, however, very different from the DSN being outlined here. Our DSN concept is one of several small arrays which individually and automatically determine arrival times and azimuths. The DSN arrays are physically independent and are separated by several kilometers. The arrival times and azimuths are communicated between sites. Such measurements from more than one node are used for weapon location but the location operation is distributed and not associated with any single site. The current military system deploys several microphones in a surveyed line with kilometer range spacings between them. All raw sensor data are forwarded to a single analysis center where acoustic time series are manually analyzed and locations estimated from arrival time information only.

The utility of seismometers is less clear at this time and will not be discussed here except to note that it is the subject of ongoing research under a separate contract independent of the DSN effort. Seismics could contribute significantly to capability but because of the limited state of current knowledge we emphasize acoustics in the remaining discussion.



NODE	COMMUNICATION/AUTO-RANGING NOMINAL CONNECTIVITY	
	5-km RANGE	10-km RANGE
1	2	4
2	4	6
3	4	8
4	4	8
5	4	6
6	2	4
7	3	5
8	4	7
9	4	8
10	4	7
11	3	5

Fig. II-20. Connectivity of 11-node DSN shown in Fig. II-19.

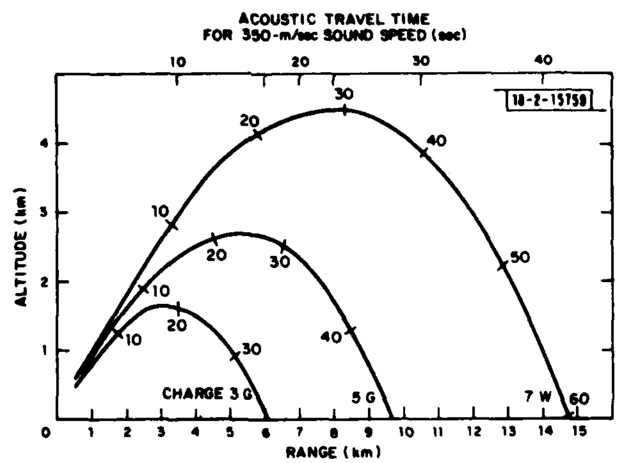


Fig. II-21. Maximum range trajectories for 155-mm Howitzer. (Adapted from Firing Tables FT 155-Q-4-1968, U.S. Government Printing Office, 1968.) Ticks show time along trajectory in 10-sec increments; top axis shows range converted to nominal acoustic travel time.

The DSN represents a system which is very different from any existing acoustic or radar systems. The following describes some of the unique or unusual characteristics of the DSN approach to the weapon location problem. These are:

(1) Areawide Availability of Information

Users need not be located at one or a few limited positions. Packet radio will allow a user within packet radio range of a DSN node to get complete and up-to-date information.

(2) No Single-Element System Failure Modes

The distributed nature of the DSN, with no critical centralized functions, makes it unusually difficult to neutralize. The two projectile radars are not an exception because the system can operate without them and, as suggested below, it may be possible to use the system to increase radar survivability from what it would be for autonomous radars.

(3) Acoustic Cueing of Radars

It is desirable to minimize the time during which radars are in use in order to maximize their survivability. Some cueing of radars from acoustic locations may be possible. To see this refer to Fig. II-21. The radars locate weapons by projectile trajectory estimation. Measurements are made during upleg, downleg, or both parts of the trajectory. It appears feasible to acoustically cue the radars in some situations, if downleg tracking is sufficient. Consider a 155-mm Howitzer firing from 5 km behind the FEBA into an area 5 km beyond the FEBA. The weapon acoustic signal will reach acoustic sensors 1 km from the FEBA in about 6/0.350 = 17 sec. Suppose it takes as long as 10 sec to process and distribute the weapon location so that the radar could be activated for projectile tracking within 27 sec of firing. From the 5G trajectory of the figure we see this is soon enough to do downleg tracking of the projectile. In general, acoustic cueing for upleg tracking would be possible only for weapons located near the FEBA firing long distances beyond the FEBA.

(4) Other Coordinated Use of Multiple Sensors and Sensor Types

Acoustic cueing of radars is a special case of sensor coordination and cooperation. There are other possibilities which could be exploited. The radars may have overlapping coverage areas. This can be exploited to improve weapon location for projectiles visible to both radars. Alternatively, the system could allocate nonredundant radar coverage, reduce radiation, and thus improve survivability. More generally, the acoustic system could be made completely responsible for some sectors to further reduce radar emissions. Finally, selected radar determined locations could be used to help make acoustic meteorological corrections. With those corrections there may be less overall need for radar locations.

(5) Self Location and Direction of Deployment

The use of electronic ranging between nodes for estimating node locations was mentioned above. This substantially increases system deployability. Moreover, since this can be done essentially in real time, the system can monitor actual deployment and, via packet radio, issue commands to correct for any errors in location. Also, situations where communication connectivity for ranging is poor and node location estimation capability is reduced can be identified and corrected during deployment. When the system or some of the nodes must be moved the system can be used to evaluate possible new locations or suggest new configurations. The evaluation would be based upon weapon-location capability, self-location capability, and communication connectivity. Connectivity might be predicted using digitized maps stored in the system.

(6) Automatic Adaptation to Number and Location of Nodes

Neither the number or location of nodes in the system is fixed. The system automatically makes the best use of the deployed configuration. It may also suppress some redundant internode communications to lower its electronic profile. For example, parts of the system may be kept totally passive until loss of other nodes or other conditions indicate substantial system benefit can be gained by breaking silence.

III. SOFTWARE DESIGN AND DEVELOPMENT

Progress in the area of software design and development is described in this section. A first-version design of the software for DSN nodes is described in Sec. A. This design is appropriate for the strawman system described in our previous SATS and for a three-node system with acoustic sensors which is being planned for FY 80. Section B reviews progress on a software test bed and general software tools required for continuing DSN research and system development. Finally, Sec. C describes a simulation package which will allow us to realistically simulate acoustic DSNs without using real data or very large amounts of computer time to simulate the computationally intensive signal processing at each node.

A. DSN NODE SOFTWARE

A structure and design for DSN node software is described in this section. The software described here is appropriate for the strawman system described in our previous SATS and is also intended as the basis for a three-node experimental system to be developed during FY 80. This software provides only lower-level DSN display and control functions. More sophisticated capability can be added as our knowledge increases.

Pictorial representations of the DSN node software are shown in Figs. III-1 and -2. Different parts of the DSN software have been assigned different areas in the figures. The width of the boundary between different software elements is intended to represent the difficulty of isolating those elements from each other for the purpose of software development. Generally speaking, software elements connected by only very narrow necks can be easily isolated from each other and be written by different teams or programmers. We should note that the ability to isolate the parts is not absolute but represents to some extent our understanding of the problem and software. For example, in the application system the boundary between sensor data reduction and detection/tracking is narrow because this boundary is relatively well understood. On the other hand, with our current state of knowledge, we do not know how to cut the detection/tracking module into smaller parts without perhaps compromising system performance. Of course, as we learn more about DSN software, we will find more narrow connections, and break the system into more pieces.

1. Application System Components

Following is a brief description of the applications software components shown in Fig. III-1.

a. Data Collection

The data collection module controls the real time collection of data from sensors and passes large buffers of data to the signal-processing module.

b. Signal Processing

The signal-processing module operates upon the raw data to obtain power-level maps and similar output, which are passed via large buffers to the data reduction module. The first versions of this will perform either normal or high-resolution frequency-wavenumber analysis and will make use of a signal-processing array processor as well as a minicomputer.

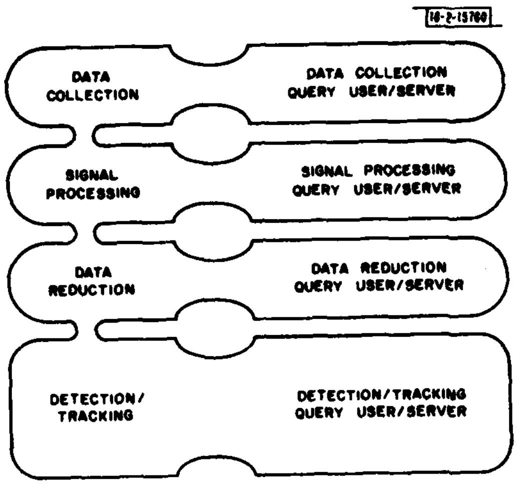


Fig. III-1. Strawman DSN node application software organization.

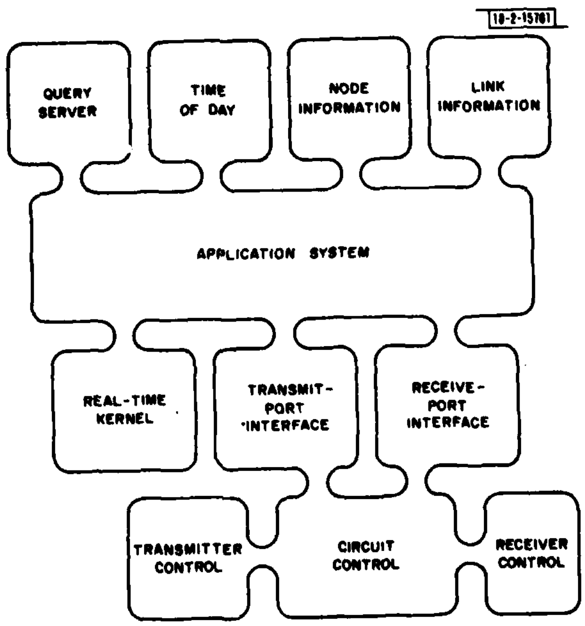


Fig. III-2. Strawman DSN node operating system software organization and relation to application software.

c. Data Reduction

The data reduction procedure involves search of power-level maps for peaks, and outputs lists of peaks with their size and shape. The output of this module goes to the detection/tracking module.

d. Detection/Tracking

This module receives power-level peak lists from the data reduction module operating at the same node. It rebroadcasts some of this sensor data to its neighbors. From its neighbors it receives sensor data from similar broadcasts, which it combines with its own sensor data to declare and track targets. It broadcasts reports containing the locations of targets, and reduced sensor data related to the located targets. Reports of this kind received from neighboring nodes are also used to help build the report broadcast by this node. Reports of targets outside the sensor range of this node and its neighbors may be rebroadcast by this node as a means of communicating such reports through the network.

The operational heart of the DSN is in this detection/tracking module, and we expect a number of rather different versions to be written over the next few years. The detection/tracking problem is difficult and first versions of this software will necessarily be limited in performance. In the longer term this module will undoubtedly be split into more parts and will incorporate and make use of more and more higher-level knowledge. It is a major focus of future DSN research.

e. Query Users and Servers

Almost every DSN module has an associated query user/server module consisting of user and server programs. The server programs run in background on the various nodes and answer questions about the data structures built by their associated module. These programs can also adjust module parameters and make measurements of module performance. The server programs are invoked by user programs which are part of the same software module as the server, but which run on different computers as well as on the same computer as the server. Thus, a person or display-and-control program on one computer can invoke a user program which will communicate with a server in some distant node to perform a task. Query user programs are the lowest layer of the display and control programs that will run at manned computer sites controlling the DSN. These same query user programs will also be the primary tools for debugging DSN programs.

2. Operating System Components

The following is a brief description of the operating system components shown in Fig. III-2. In general, these are to provide services basic to any radio-coupled distributed computing service. However, there is one important specialization. The amount of sensor and tracking information to be communicated can be larger than the available channel capacity. Thus, the software structure allows for the organization of communication in those situations as well as allowing for operation under unstressed conditions.

To avoid confusion we have not attempted to draw all the interconnections between operating system modules shown in the figure. For example, the time of day is partly determined by communications, while communications may require the time of day for packet transmission

purposes. We have also not specified a separate query user/server module for each operating system module. Most operating system modules have their own built-in query user and server programs that are used for debugging and maintenance. Finally, we note that modules related to communications may run on the node computer, on the packet radio digital section, or be split between the two, depending upon the DSN and packet radio versions.

a. Real-Time Kernel

The real-time kernel is independent of the rest of the system modules and supplies services normally associated with a diskless real-time operating system. It defines memory segments, processes, message queues, and virtual devices. It manages memory, schedules processes, moves messages and data between processes, and moves data between processes and devices.

b. Transmitter Control

This module controls the transmitter side of the radio. It accepts transmit packets, each parameterized by such things as a time to transmit, a chip code, a data rate, and a transmission power. It queues these packets and transmits them as directed by the parameters.

c. Receiver Control

This module controls the receiver side of the radio. It accepts time-stamped chip code parameters that specify what chip code to listen for at what time. It receives packets, time stamps them, and puts them on a received packet queue.

d. Circuit Control

This module manages radio circuits. The available communication capacity is in general broken into units and allocated to different DSN circuits.

For the strawman system described in the previous SATS, the circuit control module would supervise the slicing of time into different circuits, the allocation of circuit time to different nodes in a given circuit, and the assignment of chip codes to transmission times. These functions would be performed by all the nodes in a local area cooperating together.

Alternatively, the allocation could be less structured. For example, circuit transmit time might not be cooperatively assigned to avoid mutual interference but the maximum transmit bits in a second for one circuit would be limited by mutual agreement. Random access in this case would give acceptable performance as long as total rate did not become too large with respect to channel capacity.

Depending upon the manner in which the capacity is allocated, the nodes can exchange information about communications and adjust allocations accordingly. Initial DSN versions will likely have fixed allocations.

e. Receive-Port Interface

This module takes received packets from the receiver control module's queue and reconstructs messages which it then queues on receive ports. For our strawman system a packet can contain several messages so that reconstruction in that case involves breaking up the packet, not assembling long messages from multiple packets. A receive port is an interface to an application program, much like an open input file. Placement of messages into previously empty receive ports may generate application program traps.

f. Transmit-Port Interface

This module accepts messages to be transmitted and queues them on transmit ports. It then assembles packets to be transmitted from the queued transmit messages.

For our strawman system each port is also assigned to a circuit and given a priority level. When the time comes to assemble a packet for a given circuit, messages are taken from ports for that circuit. Messages from higher priority ports get preference. However, each port is limited in the number of bits worth of messages it can transmit in the last N circuit packets, where N may equal 4, 8, and 16. These limits are parameters associated with the transmit ports. Each port keeps a short history of how many bits worth of messages it has transmitted in the last several packets.

g. Query Server

The query server module maintains a continuous query server process with a query server reception port. Commands may be sent to a node's query server, which will start a process designated by the command, and pass command parameters to that process. Application query server processes can be started by this mechanism.

h. Time of Day

This module maintains an estimate of the time of day. Estimated time-of-day information is derived from a combination of internal clocks and external communications.

i. Node Information

This module maintains information about all nodes of interest to a given node, including all its neighbors. Such information as node latitude, longitude, elevation, time of day estimated by the node, and last time the node was heard from, is maintained.

j. Link Information

This module maintains information about each communication link between the node and its neighbors. Such things as estimated probability of successful transmission at two available packet radio data rates are maintained.

B. SOFTWARE TEST BED AND SOFTWARE TOOLS

The DSN node software discussed above in Sec. A will be used in our strawman low-flying aircraft surveillance system and in the first multiple-node experimental system to be developed during FY 80. It is intended that the software be developed and partially debugged and tested on simulated DSN hardware. Work is progressing on the design of the hardware simulator and flexible software tools for the development of practical DSN node software.

1. DSN Hardware Simulator Structure

Figure III-3 shows the overall structure of a DSN simulator being designed to support development and test of actual experimental DSN node software. The basic idea is to simulate hardware, including communications, and to run actual DSN node software in separate processes on a single computer. The simulator will be run on a PDP-11/70 under the UNIX operating

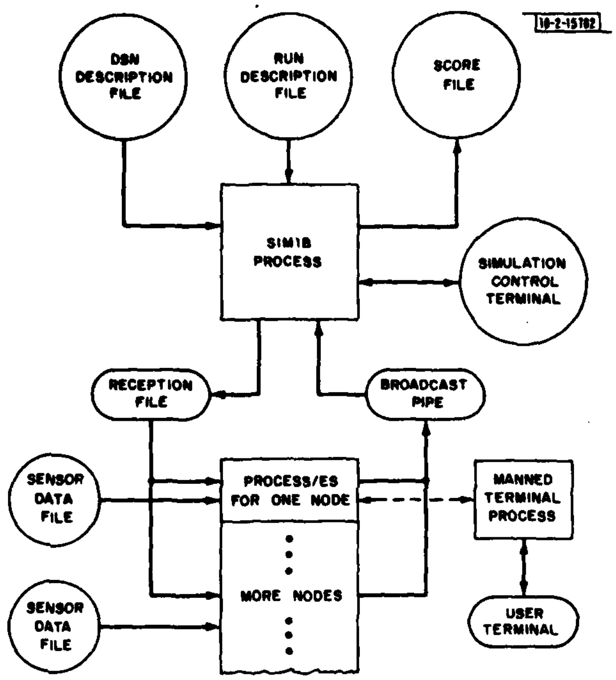


Fig. III-3. Structure of the DSN node software development simulation facility.

system. In the figure all the elements from the "Reception File" and "Broadcast Pipe" level and up represent the simulated parts of the DSN. The lower part represents the actual DSN software which will be run as separate processes. Minor exceptions apply to sensor data and the signal processing of such data. Sensor data may be input from precalculated files as shown or may be simulated through subroutines in the node processes. Furthermore, the subroutines, as discussed in Sec. C below, may not provide just raw data but may simulate outputs after the intense time series signal-processing stage.

As shown in the figure, the simulation will have a master process, the SIM1B process. It accepts formal DSN descriptions and run descriptions which it uses to cause the proper simulation to take place. It will produce a score file which can be used to study results after simulations are completed. It is the master process which initializes and starts all the DSN node processes.

There will be a large number of tasks running to perform a simulation. As such the simulation cannot be run in real time. Each event, such as a sensor input or a message transmission, will be tagged with the simulated time at which it occurs. Then the running of the tasks will be keyed to this simulated time. In this way the simulation can be run in pseudo-time while maintaining event synchronism.

The present plan is that the running of the tasks will be controlled by the master program. For efficiency, the plan is to run the tasks in a pseudo-time sequency, rather than in a time-sliced mode. When a task is run, it will work on all data accumulated, emit time-stamped

messages, and will only stop when it needs more data. It then informs the master of the next pseudo-time at which it expects to run. This approach is to help overcome swapping problems. By making sure that a task runs for as long as possible once in core, the swapping will be minimized, as will be the total time to run a simulation.

With this approach we hope to be able to develop and test DSN software for systems with perhaps as many as 10 nodes, although simulations of that size may be very slow. Some thought is being given to a much faster option which would perform the entire simulation within a single process. That would be much faster with the potential to study much larger DSNs but the size of each node process would be very much limited.

2. DSN Subroutine Library and Standard Aggregate Objects

Our current DSN research, including our new experimental data acquisition facility and our planned three-node experiments for FY 80, depend heavily upon the PDP-11 family of computers and the C programming language. It has been clear for some time that the basic software tools need to be augmented by a general-purpose set of subroutines to handle aggregates of data, such as matrices, and by the development of appropriate aggregate data objects for DSN tasks. In March 1978, a fairly successful initial set of such routines, called the SCIPL package, was written for the find-plane program that made the computations for Sec. III-B of the March 1978 SATS. Some of these routines were modified and used as a basis for the computational subroutine package described in Sec. V of the September 1978 SATS. In both cases the SCIPL routines contributed greatly to the ease of organizing program code. Based on these results we are developing a more permanent version of SCIPL.

SCIPL stands for the Scientific Programming Language, and it has the same major goal as APL: to allow subroutines to operate easily on aggregate data objects, such as matrices and tables. The major differences between SCIPL and APL are:

(a) SCIPL is compiled, not interpreted.

(b) SCIPL supports structures as well as arrays.

(c) SCIPL supports fast real-time and system programming, as well as scientific programming.

(d) SCIPL is NOT a new language by itself, but rather a collection of subroutines and macros in an existing system implementation language (C).

The first-pass design of our more permanent SCIPL is to be completed in April, and coding is to begin the same month. Major features of SCIPL are the array structures which are generalizations of those described in Sec. V-B-1 of the last SATS, and the input/output routines designed to work on aggregate objects.

In conjunction with SCIPL development as a software tool, we are in the process of defining a number of standard DSN object types, e.g., nodes and communication links. For each object we must define a structure representing the object in core memory, an ASCII text string representing the object externally, input/output subroutines for the object, and functions appropriate to the object. Table III-1 lists a few operating system objects and some of their components, by way of example.

TABLE III-1

EXAMPLES OF DSN SYSTEM OBJECTS

Object Name	Object Components	Component Units	Component Type
NODE	name		14 characters
	latitude	degrees	real
	longitude	degrees	real
	elevation	feet	real
	time estimate relative to our clock	seconds	real
	time last heard from	seconds	real
	link queue header		link pointer
LINK	link queue element 1		link pointer
	other node 1		node pointer
	link queue element 2		link pointer
	other node 2		node pointer
	time of last reception	seconds	real
	transmission success probability for two data rates	fraction	2 reals
TIME	current estimated time	seconds	real
	history of last 4 time changes, one entry per change; each entry indicates:		
	time of change	seconds	real
	amount of change	seconds	real

C. POST SIGNAL-PROCESSING MODEL OF AN ACOUSTIC ARRAY

A group of subroutines is being developed to permit a user to easily simulate the output results from processing the data from one or more small acoustic arrays: power as a function of azimuth (actually wavenumber assuming propagation parallel to the plane of the array) and frequency. These subroutines will be used to simulate the output of the DSN node signal-processing module discussed in Sec. A. They do not require real data, data-collection software, or actual signal-processing software. Most importantly, they will allow simulation of acoustic DSNs without a very large signal-processing load. The subroutines also provide a simple package that can be made available to non-Lincoln users who wish to investigate target detection and track estimation with realistic acoustic sensor inputs.

The output of the subroutines approximates the results of frequency-domain beamforming. Frequency-domain beamforming was described and computationally sized in the March 1978 SATS. By assuming a larger array aperture (see below), the output of the subroutines can be made to approximate the results of high resolution frequency-wavenumber analysis that was described in the September 1978 SATS.

1. Mathematical Model Description

In this section, a simple mathematical model for the frequency-domain beamforming output of an array is presented. The model is based on the fact that the beamforming response of a small array to acoustic signals propagating horizontally across the array from a particular azimuth can be closely approximated by a continuous array response pattern centered at the wavenumber of the acoustic source. The amplitude of the response must be scaled to account for the source-receiver distance. High-resolution frequency-wavenumber processing can be approximately modeled by beamforming an array with larger aperture.

A small acoustic array is modeled here as a circular disk of radius, R, with continuous and uniform distribution of sensors. The discussion is limited to two dimensions. The function over which one sums to derive the array response is

$$h(x,y,t) = \begin{cases} \delta(t) & \text{if } x^2 + y^2 < R^2 \\ 0 & \text{elsewhere} \end{cases} .$$

It can be shown that the response pattern as a function of wavenumber and frequency (Fourier transform) of h(x, y, t) is

$$H(k,f) = \frac{J_1(2\pi Rk)}{2\pi Rk} ,$$

where J_1 is a first-order Bessel function of the first kind and $k^2 = k_x^2 + k_y^2$. The power density spectrum function is

$$P(k,f) = H(k,f)\, H^*(k,f) = \left| \frac{J_1(2\pi Rk)}{2\pi Rk} \right|^2 ,$$

where the asterisk indicates complex conjugate. Thus, power density is a function of wavenumber (one over wavelength) and the aperture of the array and is not an explicit function of frequency.

Propagation of acoustic signals from the source or target to the array can be modeled by spherical spreading; thus, the power decreases by 20 $\log_{10}(r)$ for a distance, r. Assuming horizontal propagation across the array, the response in frequency-wavenumber space occurs on a circle defined by $k_x^2 + k_y^2 = (2\pi f/c)^2$, where c is the speed of sound in air. Steering the array or forming a beam simply translates the response in wavenumber space.

Thus, the response as a function of wavenumber along the circle defined by the speed of sound at a frequency, f, for an acoustic signal at a distance, r, from the array is

$$\left| \frac{J_1(2\pi R|\bar{k} - \bar{k}_o|)}{2\pi R|\bar{k} - \bar{k}_o|} \right|^2 \frac{T(f)}{r^2} ,$$

where $\bar{k}_o$ is the wavenumber representing the acoustic azimuth and T(f) is the power spectral density (appropriately shifted by the doppler frequency) measured at 1 m. If the circle is cut and straightened, the result is power as a function of acoustic azimuth for frequency, f. Such functions are the output of the software described below. Several such functions are simply added together for multiple targets.

2. Subroutine Package and Examples of Output

We briefly describe the abilities of the subroutines in the small acoustic array package as it now exists. Most parameters are input from the terminal. Data are stored and exchanged among the routines in files. The programs can easily be modified to also accept input from files.

a. Trajectory Input

The user can create an arbitrary 200-sec trajectory in x-, y-, and z-space. The user inputs initial target location and velocity and acceleration components. The user can change the acceleration components at any time during the 200-sec trajectory. The output file consists of a series of coordinates (x, y, and z) and velocity components for each second. The trajectory can be plotted on the graphics display.

b. Acoustic Azimuth

The user inputs the location of the small array (in the same coordinates and units in which the trajectory parameters were input) and the name of the trajectory file. The acoustic travel time and acoustic azimuth measured as a function of time at the array are computed and output to a file. A plot of acoustic azimuth vs time can be graphically displayed.

c. Target Spectrum at an Array

The source power spectral density of a target is represented as a weighted sum of shifted Gaussians as it was in the more complete model of an acoustic node which was presented in our September 1978 SATS. The amplitude, mean, and variance of each of the 5 Gaussians are specified by the user. Then propagation through the atmosphere is simply modeled by spherical spreading and the target spectrum as seen by the array model is doppler shifted in frequency by an amount appropriate to the target's radial velocity with respect to the array location.

d. Power Azimuth

At a time and frequency specified by the user, a plot of power vs azimuth is developed using the results of the first three subroutines. The plot is output to a file and graphically displayed. Techniques of including simple models of noise are currently being investigated and have not yet been included in the subroutine.

An example of a power plot from this software is presented in Fig. III-4. For this figure, the frequency was set to 350 Hz and the array aperture to R = 1.5 m. Arrows show the true azimuths of sound arrival and they correspond to the peaks of power. However, when this subroutine is augmented with noise, it is not difficult to imagine confusing target peaks with sidelobe features. The situation might be better with high-resolution analysis which is adaptive and often discriminates against sidelobes.

Figure III-5 is a similar plot with an assumed frequency of analysis of 150 Hz. There appears to be a major peak between 180° and 200° azimuth. In fact, there are two sources with sound arriving from the azimuth indicated by the arrows. When the assumed frequency of analysis is increased to 450 Hz, or equivalently the radius of the array is increased proportionally, the results are as plotted in Fig. III-6. The broad peak of Fig. III-5 has been resolved into two peaks, one at 160° and the other at 200°.

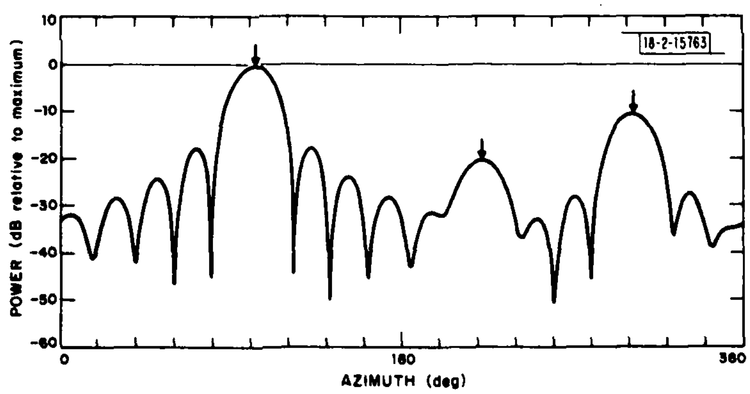


Fig. III-4. Power as a function of acoustic azimuth. Analysis frequency is 350 Hz. Peaks at 100°, 220°, and 300°. Arrows indicate true azimuths of three targets.

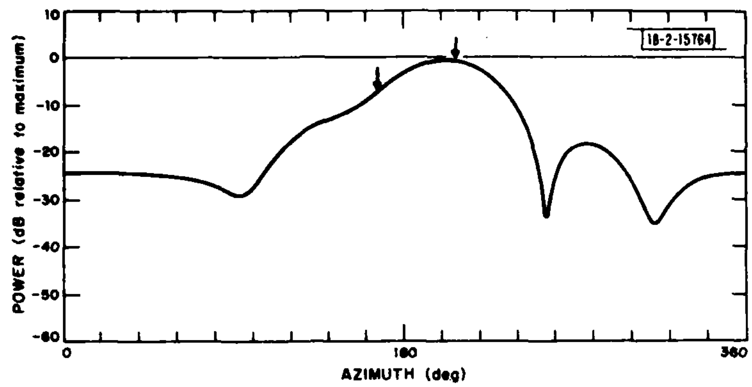


Fig. III-5. Simulated output of power vs acoustic azimuth. Analysis frequency is 150 Hz. Arrows indicate true azimuths of two targets.

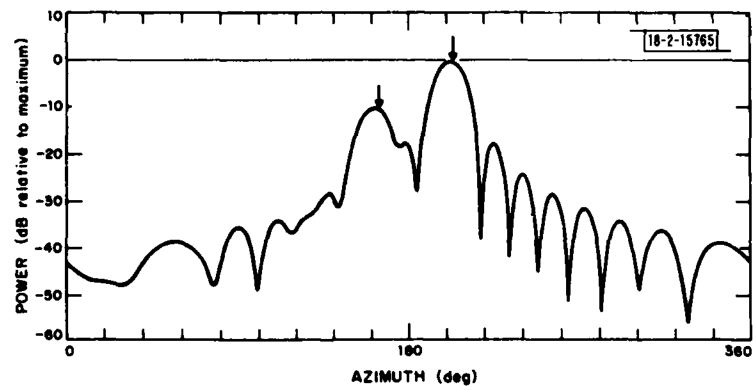


Fig. III-6. Simulated output of power vs acoustic azimuth. Analysis frequency is 450 Hz. Arrows indicate true azimuth of the targets.

IV. EXPERIMENTAL FACILITIES

One of the goals of the DSN project for FY 79 is to begin to augment the acoustic array experimental data base that was initiated with data from the Ft. Huachuca experiments with data collected in different climatic and noise environments. Considerable progress has been made in designing and selecting components for the digital data acquisition system that will be used to record more acoustic array data. The requirements and preliminary specifications for such a system were presented in the September 1978 SATS.

In summary, high-sensitivity, large dynamic range microphones will be used in the array. The electrical signals from each microphone will be preamplified and appropriately filtered. The signals are then multiplexed and input to an A/D system with an automatic gain-ranging amplifier. The resulting digital output consists of 13 to 14 bits of A/D conversion and 2 to 3 bits that represent gain information. The digital signals are input to a minicomputer where they are formatted for recording onto magnetic tape. The recorded data can then be transferred to facilities for off-line processing.

A PDP-11/34 processor with 64K of MOS memory has been purchased for the data acquisition system. The minicomputer will not only transfer data from the A/D converter to magnetic tape, but also will provide techniques for maintaining system calibration and will provide the user with real-time status and displays of the data being recorded. The major reason for selecting the PDP-11/34 is its hardware and software compatibility with the PDP-11/70 that is currently being used under the UNIX operating system for the DSN program. Data acquisition software will be developed in the C programming language on the larger facility (work can actually begin before the arrival of the new processor) and can be directly transferred to the 11/34. Peripheral devices (for example, the magnetic tape drives) can also be interfaced to the current facilities, checked out, and then easily interfaced to the 11/34.

The two PDP-11's will be connected by synchronous communication links. The Digital DMC11 and the ACC UMC Z-80 are currently being investigated for this purpose. With such devices, the 11/34 can be down-loaded and the data acquisition software started remotely. The link can be a cable within 2000 ft of the 11/70 or a telephone line with modems for longer distances.

Specifications were defined for the automatic gain-ranging A/D system and the magnetic tape recording system. Requests for quotations (RFQs) were issued to vendors and the responses are currently being evaluated.

A brief summary of the A/D conversion system specifications is presented below:

- 8 to 16 channels, each one with 2000-Hz sampling.
- 32-kHz, minimum system throughput.
- 14-bit A/D conversion with gains of 1, 16, and 64 or equivalent that results in approximately 20 bits of resolution and range.
- DMA interface to the PDP-11 Unibus.

The following options were included in the specifications:

- Switchable sampling rates (500, 1000, 2000, and 4000 Hz).
- Low-noise preamplifiers (×100) and anti-aliasing filters.
- Calibration hardware.
- Battery operation.

The specification for two magnetic tape drives and a controller listed 9-track, dual-density (800/1600 bytes per inch) drives with tape speed of at least 45 in./sec. The controller must be completely compatible with the PDP-11/34.

All components are being selected keeping in mind that the complete data acquisition system will eventually be required to operate in the field from battery power. Furthermore, since the sensors are microphones, the equipment must operate with minimum acoustical noise.

It is expected that assembly of the data acquisition system will begin as soon as components are delivered. Data acquisition software must be designed and implemented.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-79-211 ✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Distributed Sensor Networks		5. TYPE OF REPORT & PERIOD COVERED Semiannual Technical Summary 1 October 1978 – 31 March 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard T. Lacoss		8. CONTRACT OR GRANT NUMBER(s) F19628-78-C-0002 ✓
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. / P.O. Box 73 Lexington, MA 02173		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order 3345 Program Element No. 61101E Project No. 9D30
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209		12. REPORT DATE 31 March 1979
		13. NUMBER OF PAGES 60
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB Bedford, MA 01731		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

multiple-sensor surveillance system
multisite detection
target surveillance and tracking
2-dimensional search-space cell
acoustic, seismic, radar sensors
low-flying aircraft
packet radio
high-resolution search-algorithms

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Progress on the Distributed Sensor Networks program is reported. Further design and analysis of strawman sensor and report circuit communications for a low-flying aircraft DSN are presented. System studies dealing with sensor redundancy, adaptive communications, nonuniform node distribution, and alternative DSN applications are summarized. Overall designs for DSN node software, for a software development DSN simulation test bed, and acoustic node front-end simulation software are reviewed. Progress on development of a digital acoustic data collection system is reported.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)